



MATLAB PROGRAMS LISTING



Example	Title	Page
2.8	Difference-equation solution	38
2.12	Partial-fraction expansion	44
2.15	Discrete convolution	47
2.22	Characteristic values and vectors	67
2.24	Transfer function from state equations	70
2.25	Solution of discrete state equations	72
Pr. 2-21	Digital controller simulation	84
4.14	Discrete model from analog model	160
6.4	Step response of a sampled-data system	207
7.7	Root locus plot	249
7.12	Bode diagram plot	262
8.2	Digital controller design	302
8.4	Digital controller design	317
9.2	Pole assignment design	344
9.3	Prediction observer design	350
9.7	Reduced-order observer design	359
9.9	Current observer design	363
10.2	Linear quadratic optimal design	394
10.6	Least-squares system identification	409
10.8	Kalman filter design	416
11.6	Butterworth filter design	447
11.7	Chebyshev filter design	452

Digital Control System Analysis and Design

Third Edition

CHARLES L. PHILLIPS

*Department of Electrical Engineering
Auburn University*

H. TROY NAGLE

*Department of Electrical
and Computer Engineering
North Carolina State University*



PRENTICE HALL, Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data

Phillips, Charles L.

Digital control system analysis and design / Charles L. Phillips,
H. Troy Nagle. -- 3rd ed.

p. cm.

Includes bibliographical references and index.

ISBN: 0-13-309832-X

1. Digital control systems. 2. Electric filters, Digital.

3. Intel 8086 (Microprocessort) 4. MATLAB. I. Nagle, H. Troy
1942- . II. Title.

TJ223.M53P47 1995

629.8'95--dc20

94-3482

CIP

Acquisitions editor: **LINDA RATTS-ENGELMAN**

Production editor: **RICHARD DeLORENZO**

Copy editor: **BARBARA ZEIDERS**

Cover designer: **WENDY ALLING JUDY**

Prepress buyer: **LORI BULWIN**

Manufacturing buyer: **BILL SCAZZERO**

Editorial assistant: **NAOMI GOLDMAN**



©1995 by Prentice-Hall, Inc.

A Simon & Schuster Company

Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book may be
reproduced, in any form or by any means,
without permission in writing from the publisher.

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

MATLAB is a registered trademark of The MathWorks, Inc.

Printed in the United States of America

10 9 8 7 6

ISBN 0-13-309832-X

Prentice-Hall International (UK) Limited, London

Prentice-Hall of Australia Pty. Limited, Sydney

Prentice-Hall Canada Inc., Toronto

Prentice-Hall Hispanoamericana, S.A., Mexico

Prentice-Hall of India Private Limited, New Delhi

Prentice-Hall of Japan, Inc., Tokyo

Simon & Schuster Asia Pte. Ltd., Singapore

Editora Prentice-Hall do Brasil, Ltda., Rio de Janeiro

To
Laverne, Susie, Chuck, and Carole
Julia

Contents

PREFACE	xi
PREFACE TO COMPUTER-AIDED ANALYSIS AND DESIGN PROGRAMS	xv
1 INTRODUCTION	1
1.1 Overview	1
1.2 Digital Control System	3
1.3 The Control Problem	7
1.4 Satellite Model	9
1.5 Servomotor System Model	10
1.6 Temperature Control System	16
1.7 Summary	18
References	18
Problems	19
2 DISCRETE-TIME SYSTEMS AND THE z-TRANSFORM	27
2.1 Introduction	27
2.2 Discrete-Time Systems	27
2.3 Transform Methods	30
2.4 Properties of the z-Transform	31
2.5 Solution of Difference Equations	37

2.6	The Inverse z -Transform	40
2.7	Simulation Diagrams and Flow Graphs,	48
2.8	State Variables	53
2.9	Other State-Variable Formulations	62
2.10	Transfer Functions	68
2.11	Solutions of the State Equations	71
2.12	Linear Time-Varying Systems	76
2.13	Summary	77
	References and Further Reading	77
	Problems	78

3 SAMPLING AND RECONSTRUCTION

89

3.1	Introduction	89
3.2	Sampled-Data Control Systems	89
3.3	The Ideal Sampler	92
3.4	Evaluation of $E^*(s)$	95
3.5	Results from the Fourier Transform	97
3.6	Properties of $E^*(s)$	99
3.7	Data Reconstruction	102
3.8	Digital-to-Analog Conversion	111
3.9	Analog-to-Digital Conversion	113
3.10	Summary	124
	References and Further Reading	125
	Problems	125

4 OPEN-LOOP DISCRETE-TIME SYSTEMS

131

4.1	Introduction	131
4.2	The Relationship between $E(z)$ and $E^*(s)$	132
4.3	The Pulse Transfer Function	133
4.4	Open-Loop Systems Containing Digital Filters	138
4.5	The Modified z -Transform	142
4.6	Systems with Time Delays	144
4.7	Nonsynchronous Sampling	147
4.8	State-Variable Models	150
4.9	Review of Continuous State Variables	152
4.10	Discrete State Equations	156
4.11	Practical Calculations	159
4.12	Summary	161
	References and Further Reading	161
	Problems	162

5 CLOSED-LOOP SYSTEMS**173**

- 5.1 Introduction 173
- 5.2 Preliminary Concepts 173
- 5.3 Derivation Procedure 176
- 5.4 State-Variable Models 183
- 5.5 Summary 191
- References and Further Reading 192
- Problems 192

6 SYSTEM TIME-RESPONSE CHARACTERISTICS**202**

- 6.1 Introduction 202
- 6.2 System Time Response 202
- 6.3 System Characteristic Equation 210
- 6.4 Mapping the s-Plane into the z-Plane 210
- 6.5 Steady-State Accuracy 218
- 6.6 Simulation 221
- 6.7 Control Software 226
- 6.8 Summary 227
- References and Further Reading 227
- Problems 227

7 STABILITY ANALYSIS TECHNIQUES**235**

- 7.1 Introduction 235
- 7.2 Stability 235
- 7.3 Bilinear Transformation 240
- 7.4 The Routh–Hurwitz Criterion 242
- 7.5 Jury's Stability Test 245
- 7.6 Root Locus 249
- 7.7 The Nyquist Criterion 252
- 7.8 The Bode Diagram 261
- 7.9 Interpretation of the Frequency Response 264
- 7.10 Closed-Loop Frequency Response 266
- 7.11 Summary 271
- References and Further Reading 271
- Problems 272

8 DIGITAL CONTROLLER DESIGN**281**

- 8.1 Introduction 281
- 8.2 Control System Specifications 282
- 8.3 Compensation 289

8.4	Phase-Lag Compensation	291
8.5	Phase-Lead Compensation	297
8.6	Phase-Lead Design Procedure	300
8.7	Lag-Lead Compensation	307
8.8	Integration and Differentiation Filters	310
8.9	PID Controllers	312
8.10	PID Controller Design	315
8.11	Design by Root Locus	319
8.12	Summary	327
	References and Further Reading	327
	Problems	328

9 POLE-ASSIGNMENT DESIGN AND STATE ESTIMATION

337

9.1	Introduction	337
9.2	Pole Assignment	338
9.3	State Estimation	345
9.4	Reduced-Order Observers	357
9.5	Current Observers	361
9.6	Controllability and Observability	365
9.7	Systems with Inputs	369
9.8	Summary	374
	References and Further Reading	375
	Problems	376

10 LINEAR QUADRATIC OPTIMAL CONTROL

382

10.1	Introduction	382
10.2	The Quadratic Cost Function	384
10.3	The Principle of Optimality	386
10.4	Linear Quadratic Optimal Control	389
10.5	The Minimum Principle	397
10.6	Steady-State Optimal Control	398
10.7	Least-Squares Curve Fitting	404
10.8	Least-Squares System Identification	406
10.9	Recursive Least-Squares System Identification	410
10.10	Optimal State Estimation—Kalman Filters	413
10.11	Least-Squares Minimization	420
10.12	Summary	421
	References and Further Reading	421
	Problems	422

11	SAMPLED-DATA TRANSFORMATION OF ANALOG FILTERS	430
11.1	Introduction	430
11.2	Sampled-Data Transformations	430
11.3	Review of Continuous Filter Design	447
11.4	Transforming Analog Filters	455
11.5	Summary	462
	References	462
	Problems	463
12	DIGITAL FILTER STRUCTURES	465
12.1	Introduction	465
12.2	Direct Structures	465
12.3	Second-Order Modules	470
12.4	Cascade Realization	475
12.5	Parallel Realization	478
12.6	PID Controllers	483
12.7	Ladder Realization	485
12.8	Other Structures	490
12.9	Summary	490
	References	490
	Problems	491
13	MICROCOMPUTER IMPLEMENTATION OF DIGITAL FILTERS	493
13.1	Introduction	493
13.2	The Intel 80 × 86 [1]	493
13.3	Implementing Second-Order Modules [3]	497
13.4	Parallel Implementation of Higher-Order Filters	505
13.5	Cascade Implementation of Higher-Order Filters	506
13.6	Comparison of Structures	512
13.7	LabVIEW [7,8]	512
13.8	Summary	523
	References	524
	Problems	524
14	FINITE-WORDLENGTH EFFECTS	525
14.1	Introduction	525
14.2	Fixed-Point Number Systems	525

14.3	Coefficient Quantization	541
14.4	Signal Quantization Analysis	546
14.5	Limit Cycles	561
14.6	Impact of Finite Wordlength on Filter Implementation	574
14.7	Cascaded Second-Order Modules	575
14.8	Parallel Second-Order Modules	590
14.9	Summary	593
	References	593
	Problems	595

15 CASE STUDIES

597

15.1	Introduction	597
15.2	Servomotor System	598
15.3	Environmental Chamber Control System	605
15.4	Aircraft Landing System	613
	References	622

APPENDIXES

I	Design Equations	624
II	Mason's Gain Formula	626
III	Evaluation of $E^*(s)$	631
IV	Review of Matrices	637
V	Second-Order Module Subroutines	645
VI	Control Software	654
VII	The Laplace Transform	660
VIII	z-Transform Tables	675

INDEX

679

Preface

This book is intended to be used primarily as a text for a first course in discrete control systems and/or a first course in digital filters, at either the senior or first-year graduate level. Furthermore, the text is suitable for self-study by the practicing engineer.

This book is based on material taught at both Auburn University and North Carolina State University, and in intensive short courses taught in both the United States and Europe. The practicing engineers who attended these short courses have influenced both the content and the direction of this book, resulting in more emphasis placed on the practical aspects of designing and implementing digital control systems. Also, the introduction of the microprocessor has greatly influenced the material of the book, with Chapter 13 devoted exclusively to microcomputer implementations.

Chapter 1 presents a brief introduction and an outline of the text. Chapters 2 through 10 cover the analysis and design of discrete-time linear control systems. Some previous knowledge of continuous-time control system is helpful in understanding this material.

The mathematics involved in the analysis and design of discrete-time control systems is the z -transform and vector-matrix difference equations, with these topics presented in Chapter 2.

Chapter 3 is devoted to the very important topic of sampling signals; and the mathematical model of the sampler and data hold, which is basic to the remainder of the text, is developed here. The implications and the limitations of this model are stressed. In addition, analog-to-digital and digital-to-analog converters are discussed.

The next four chapters, 4, 5, 6, and 7, are devoted to the application of the mathematics of Chapter 2 to the analysis of discrete-time systems, with emphasis on digital control systems. Classical design techniques are covered in Chapter 8, with the frequency-response Bode technique emphasized. Modern design techniques are presented in Chapters 9 and 10. Throughout these chapters, practical computer-aided analysis and design are stressed.

Chapters 11, 12, 13, and 14 are devoted to digital filters. In Chapter 11 the transformation of analog filters into discrete-time representations is presented. The properties of numerical integration techniques and their relation to sampled-data transformations are investigated. Chapter 12 demonstrates various structures for digital filters. Cascade and parallel arrangements are detailed.

Implementation of digital filters on microcomputers is the subject of Chapter 13. Assembly language programs for the INTEL 80x86 and other 16-bit machines are included. Several other signal processors and microcomputers are discussed.

Chapter 14 covers many of the theoretical aspects of digital filtering. Quantization effects on signal amplitude and filter coefficients are discussed. Quantization noise is examined and characterized. Limit cycles are investigated. The theoretical aspects are then employed in practical guidelines for implementing digital filters. Presented in Chapter 15 are case studies of three operational digital control systems.

In this third edition, many of the explanations related to basic material have been clarified. A short discussion of pertinent material of the Fourier transform has been added to Chapter 3. This material aids in understanding the effects of sampling a signal. In addition, the material on root-locus design in Chapter 8 has been clarified and expanded.

Most of the end-of-chapter problems are new. Each problem has been written to illustrate basic material in the chapter. In most problems, the student is led through a second method of solution, in order to verify the results. This approach also relates different procedures to each other. As a result, the problem statements tend to be longer than in earlier editions. However, the problems are stated such that the second solution can be omitted if desired. In many problems the student is also asked to verify the results using MATLAB or either of the programs CTRL or CSP, which are written specifically for this book. (The programs CTRL and CSP are described in Appendix VI.) Generally, if applicable, short MATLAB programs are given with examples to illustrate the computer calculation of the results of the example. These programs are easily modified for the homework problems. Of course, the problem parts related to verification by computer may be omitted if desired.

At Auburn University, three courses based on the controls portion of this text, Chapters 2 through 10, have been taught. Chapters 2 through 8 are covered in their entirety in a one-quarter four-credit-hour graduate course. Thus the material is also suitable for a three-semester-hour course and has been presented as such at North Carolina State University. These chapters have also been covered in twenty lecture hours of an undergraduate course, but with much of the material omitted. The topics not covered in this abbreviated presentation are state variables, the

modified z -transform, nonsynchronous sampling, and closed-loop frequency response. A third course, which is a one-quarter three-credit course, requires one of the above courses as a prerequisite, and introduces the state variables of Chapter 2. Then the state-variable models of Chapter 4, and the modern design of Chapters 9 and 10, are covered in detail.

Also at Auburn University, a first course in digital filters has been taught using material from Chapters 2, 11, 12, 13, and 14. The course was offered to senior and beginning graduate students for three quarter hours credit and was organized around 28 lectures.

To further assist the user of this book, a manual containing problem solutions has been developed. The authors feel that the problems at the end of the chapters are an indispensable part of the text, and should be fully utilized by all who study this book.

Finally, we gratefully acknowledge the many colleagues, graduate and undergraduate students, and staff members of the Electrical Engineering Department at Auburn University who have contributed to the development of this book. In particular, we wish to thank Professor Richard C. Jaeger for contributing the digital-to-analog and analog-to-digital sections of Chapter 3. We are especially indebted to Professor J. David Irwin, Electrical Engineering Department Head at Auburn University, for his aid and encouragement.

.

:

.

Preface to Computer-Aided Analysis and Design Programs

The availability of small computers, such as the IBM PC®, has expanded the student's educational opportunity. One advantage of these computers is in the computer verification of the examples in a textbook and of the student's solutions for problems. The authors of this book, along with Professor B. Tarik Oranc, have developed two programs that run on compatible IBM PC's. The first of these programs, CTRL, is based on MATLAB®. The second program, CSP, is a compiled program and stands alone. The programs are described in Appendix VI.

Both programs are menu driven. In addition, the user is prompted at appropriate times for required data, such as transfer functions, state models, initial conditions, and so on.

CTRL is a MATLAB toolbox and requires the student version of MATLAB be resident in the computer memory. CTRL will perform almost all calculations used in the examples and problems in this book. However, no programming in MATLAB is required. This allows students to allot available time to the study of the fundamentals of digital control, rather than debugging programs. This program may be obtained without charge; see the form at the rear of this book. CTRL applies to both digital and analog control systems, and also contain some of the polynomial and matrix manipulations of MATLAB as related to control systems. However, no programming is required. CTRL may be obtained without cost (see Appendix VI).

CSP is similar to CTRL, but does not require MATLAB. CSP also applies to both analog and digital control systems. Instructors may obtain CSP without cost from the first author at: Department of Electrical Engineering, Auburn University, AL, 36849-5201. CSP may then be copied as required for educational purposes. See Appendix VI for further descriptions of these programs.

Charles L. Phillips
Auburn University

H. Troy Nagle
North Carolina State University

[Redacted header line]

Page 1 of 1

Introduction

1.1 OVERVIEW

This book is concerned with the analysis and design of closed-loop physical systems that contain digital computers. The computers are placed within the system to modify the dynamics of the closed-loop system such that a *more satisfactory* system response is obtained.

A *closed-loop system* is one in which certain system forcing functions (inputs) are determined, at least in part, by the response (outputs) of the system (i.e., the input is a function of the output). A simple closed-loop system is illustrated in Figure 1-1. The physical system (process) to be controlled is called the *plant*. Usually a system, called the *control actuator*, is required to drive the plant; in Figure 1-1 the actuator has been included in the plant. The *sensor* (or sensors) measures the response of the plant, which is then compared to the desired response. This difference signal initiates actions that result in the actual response approaching the desired response, which drives the difference signal toward zero. Generally, an unacceptable closed-loop response occurs if the plant input is simply the difference in the desired response and the actual response. Instead, this difference signal must be processed (filtered) by another physical system, which is called a *compensator*, a *controller*, or simply a *filter*. One problem of the control system designer is the design of the compensator.

An example of a closed-loop system is the case of a pilot landing an aircraft. For this example, in Figure 1-1 the plant is the aircraft and the plant inputs are the pilot's manipulations of the various control surfaces and of the aircraft velocity. The pilot is the sensor, with his or her visual perceptions of position, velocity, instrument

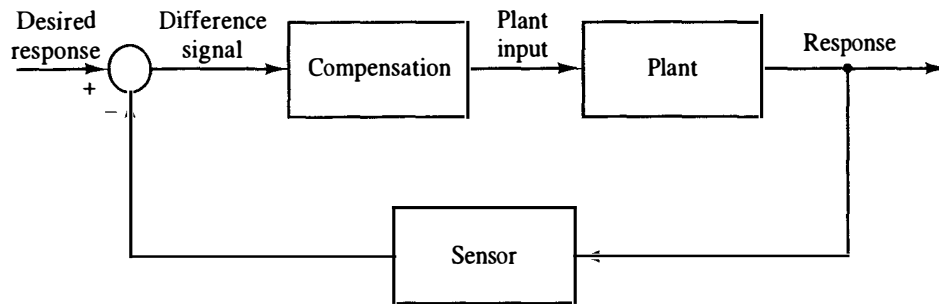


Figure 1-1 Closed-loop system.

indications, and so on, and with his or her sense of balance, motion, and so on. The desired response is the pilot's concept of the desired flight path. The compensation is the pilot's manner of correcting perceived errors in flight path. Hence, for this example, the compensation, the sensor, and the generation of the desired response are all functions performed by the pilot. It is obvious from this example that the compensation must be a function of plant (aircraft) dynamics. A pilot trained only in a fighter aircraft is not qualified to land a large passenger aircraft, even if he or she can manipulate the controls.

We will consider systems of the type shown in Figure 1-1, in which the sensor is an appropriate measuring instrument and the compensation function is performed by a digital computer. The plant has dynamics; we will program the computer such that it has *dynamics* of the same nature as those of the plant. Furthermore, although generally we cannot choose the dynamics of the plant, we can choose those of the computer such that, in some sense, the dynamics of the closed-loop system are *satisfactory*. For example, if we are designing an automatic aircraft landing system, the landing must be safe, the ride must be acceptable to the pilot and to any passengers, and the aircraft cannot be unduly stressed.

Both classical and modern control techniques of analysis and design are developed in this book. Almost all control-system techniques developed are applicable to *linear time-invariant discrete-time* system models. A linear system is one for which the principle of superposition applies [1]. Suppose that the input of a system $x_1(t)$ produces a response (output) $y_1(t)$, and the input $x_2(t)$ produces the response $y_2(t)$. Then, if the system is linear, the principle of superposition applies and the input $[a_1x_1(t) + a_2x_2(t)]$ will produce the output $[a_1y_1(t) + a_2y_2(t)]$, where a_1 and a_2 are any constants. All physical systems are inherently nonlinear; however, in many systems, if the system signals do not vary over too wide a range, the system responds in a linear manner. Even though the analysis and design techniques presented are applicable to linear systems only, certain nonlinear effects will be discussed.

When the parameters of a system are constant with respect to time, the system is called a time-invariant system. An example of a time-varying system is the booster stage of a space vehicle, in which fuel is consumed at a known rate; for this case, the mass of the vehicle decreases with time.

A *discrete-time system* has signals that can change values only at discrete

instants at time. We will refer to systems in which all signals can change continuously with time as *continuous-time*, or *analog*, systems.

The compensator, or controller, in this book is a digital filter. The filter implements a transfer function. The design of transfer functions for digital controllers is the subject of Chapters 2 through 11. Once the transfer function is known, algorithms for its realization must be programmed on a digital computer, or the algorithms must be implemented in special-purpose hardware. These subjects are detailed in Chapters 12, 13, and 14. In Chapter 15 we present three case studies of digital controls systems.

Presented next in this chapter is an example of a digital control system. Then the equations describing three typical plants that appear in closed-loop systems are developed.

1.2 DIGITAL CONTROL SYSTEM

The basic structure of a digital control system will be introduced through the example of an automatic aircraft landing system. The system to be described is similar to the landing system that is currently operational on U.S. Navy aircraft carriers [2]. Only the simpler aspects of the system will be described.

The automatic aircraft landing system is depicted in Figure 1-2. The system consists of three basic parts: the aircraft, the radar unit, and the controlling unit. During the operation of this control system, the radar unit measures the approximate vertical and lateral positions of the aircraft, which are then transmitted to the controlling unit. From these measurements, the controlling unit calculates appropriate pitch and bank commands. These commands are then transmitted to the aircraft autopilots, which in turn cause the aircraft to respond accordingly.

In Figure 1-2 the controlling unit is a digital computer. The lateral control system, which controls the lateral position of the aircraft, and the vertical control system, which controls the altitude of the aircraft, are independent (decoupled). Thus the bank command input affects only the lateral position of the aircraft, and the pitch command input affects only the altitude of the aircraft. To simplify the treatment further, only the lateral control system will be discussed.

A block diagram of the lateral control system is given in Figure 1-3. The aircraft lateral position, $y(t)$, is the lateral distance of the aircraft from the extended center-line of the runway. The control system attempts to force $y(t)$ to zero. The radar unit measures $y(t)$ every 0.05 s. Thus $y(kT)$ is the sampled value of $y(t)$, with $T = 0.05$ s and $k = 0, 1, 2, 3, \dots$. The digital controller processes these sampled values and generates the discrete bank commands $\phi(kT)$. The data hold, which is on board the aircraft, clamps the bank command $\phi(t)$ constant at the last value received until the next value is received. Then the bank command is held constant at the new value until the following value is received. Thus the bank command is updated every $T = 0.05$ s, which is called the *sample period*. The aircraft responds to the bank command, which changes the lateral position $y(t)$.



Two-crew-member airline flight deck. The digital electronics include an automatic flight control system (i.e., "automatic pilot"). (Courtesy of Boeing Airplane Company.)

Two additional inputs are shown in Figure 1-3. These are unwanted inputs, called *disturbances*, and we would prefer that they not exist. The first, $w(t)$, is the wind input, which certainly affects the position of the aircraft. The second disturbance input, labeled radar noise, is present since the radar cannot measure the exact position of the aircraft. This noise is the difference between the exact aircraft position and the measured position. Sensor noise is always present in a control system, since no sensor is perfect.

The design problem for this system is to maintain $y(t)$ small in the presence of the wind and radar-noise disturbances. In addition, the plane must respond in a manner that both is acceptable to the pilot and does not unduly stress the structure of the aircraft.

To effect the design, it is necessary to know the mathematical relationships

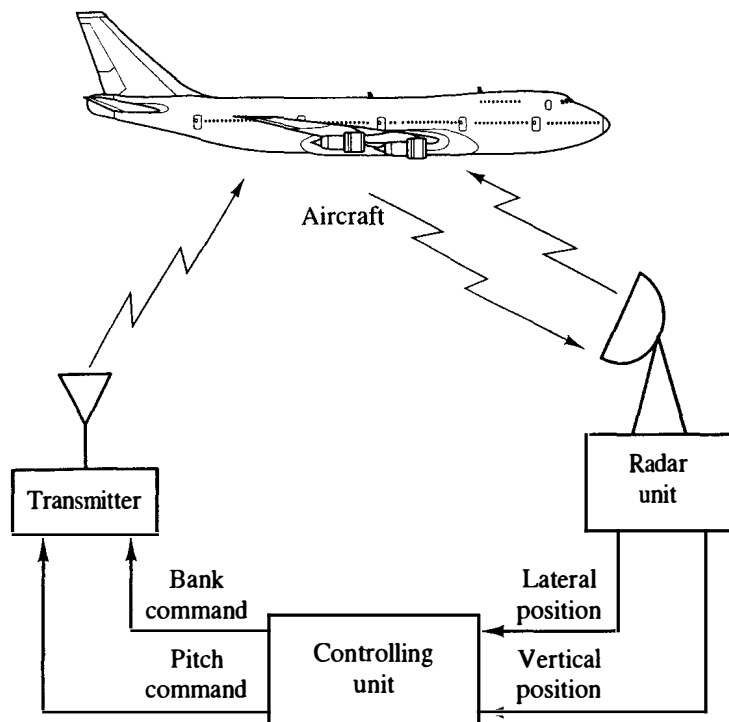


Figure 1-2 Automatic aircraft landing system.

between the wind input $w(t)$, the bank command input $\phi(t)$, and the lateral position $y(t)$. These mathematical relationships are referred to as the mathematical model, or simply the model, of the aircraft. For example, for the McDonnell-Douglas Corporation F4 aircraft, the model of lateral system is a ninth-order ordinary nonlinear differential equation [3]. For the case that the bank command $\phi(t)$ remains small in amplitude, the nonlinearities are not excited and the system model is a ninth-order ordinary linear differential equation.

The task of the control system designer is to specify the processing to be accomplished in the digital controller. This processing will be a function of the ninth-order aircraft model, the expected wind input, the radar noise, the sample period T , and the desired response characteristics. Various methods of digital controller design are developed in Chapters 8, 9, 10, and 11.

The development of the ninth-order model of the aircraft is beyond the scope of this book. In addition, this model is too complex to be used in an example in this book. Hence, to illustrate the development of models of physical systems, the mathematical models of three simple, but common, control-system plants will be developed later in this chapter. Two of the systems relate to the control of position, and the third relates to temperature control. In addition, Chapter 10 presents a procedure for determining the model of a physical system from input-output measurements on the system.

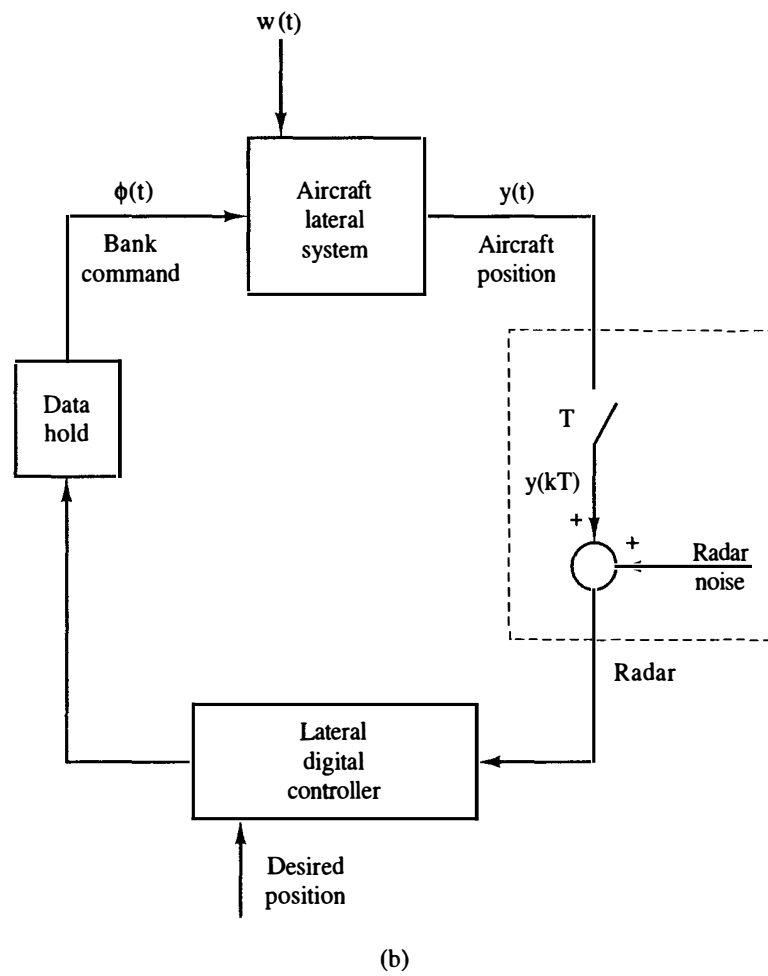
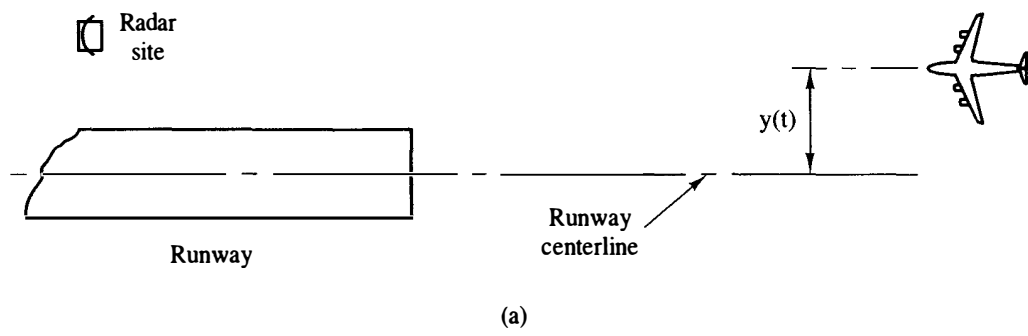


Figure 1-3 Aircraft lateral control system.

1.3 THE CONTROL PROBLEM

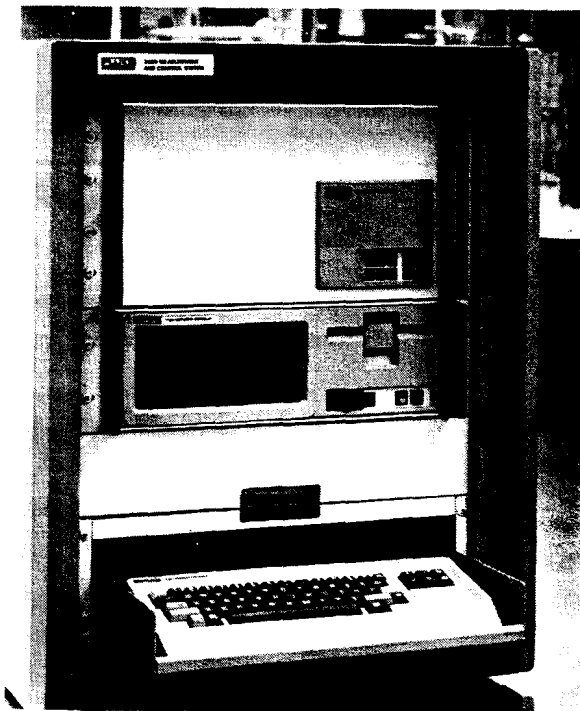
We may state the control problem as follows. A physical system or process is to be accurately controlled through closed-loop, or feedback, operation. An output variable (signal), called the response, is adjusted as required by an error signal. The error signal is a measure of the difference between the system response, as determined by a sensor, and the desired response.

Generally, a controller, or filter, is required to process the error signal in order that certain control criteria, or specifications, will be satisfied. The criteria may involve, but not be limited to:

1. Disturbance rejection
2. Steady-state errors
3. Transient response
4. Sensitivity to parameter changes in the plant

Solving the control problem will generally involve:

1. Choosing sensors to measure the required feedback signals
2. Choosing actuators to drive the plant
3. Developing the plant, sensor, and actuator models (equations)
4. Designing the controller based on the developed models and the control criteria



Microcomputer-based measurement system and digital controller. (Courtesy of John Fluke Manufacturing Company.)

5. Evaluating the design analytically, by simulation, and finally, by testing the physical system
6. Iterating this procedure until a satisfactory physical-system response results

Because of inaccuracies in the mathematical models, the initial tests on the physical system may not be satisfactory. The controls engineer must then iterate this design procedure, using *all* tools available, to improve the system. Intuition, developed while experimenting with the physical system, usually plays an important part in the design process.

Figure 1-4 illustrates the relationship of mathematical analysis and design to physical-system design procedures [4]. In this book, all phases shown in the figure are discussed, but the emphasis is necessarily on the conceptual part of the procedures—the application of mathematical concepts to mathematical models. In practical design situations, however, the major difficulties are in formulating the problem mathematically and in translating the mathematical solution back to the physical world. Many iterations of the procedures shown in Figure 1-4 are usually required in practical situations.

Depending on the system and the experience of the designer, some of the steps listed earlier may be omitted. In particular, many control systems are implemented by choosing standard forms of controllers and experimentally determining the parameters of the controller; a specified step-by-step procedure is applied directly to the physical system, and no mathematical models are developed. This type of procedure works very well for certain control systems. For other systems, it does not. For example, a control system for a space vehicle cannot be designed in this manner; this system must perform satisfactorily the first time it is activated.

In this book mathematical procedures are developed for the analysis and design of control systems. The techniques developed may or may not be of value in the design of a particular control system. However, standard controllers are utilized in

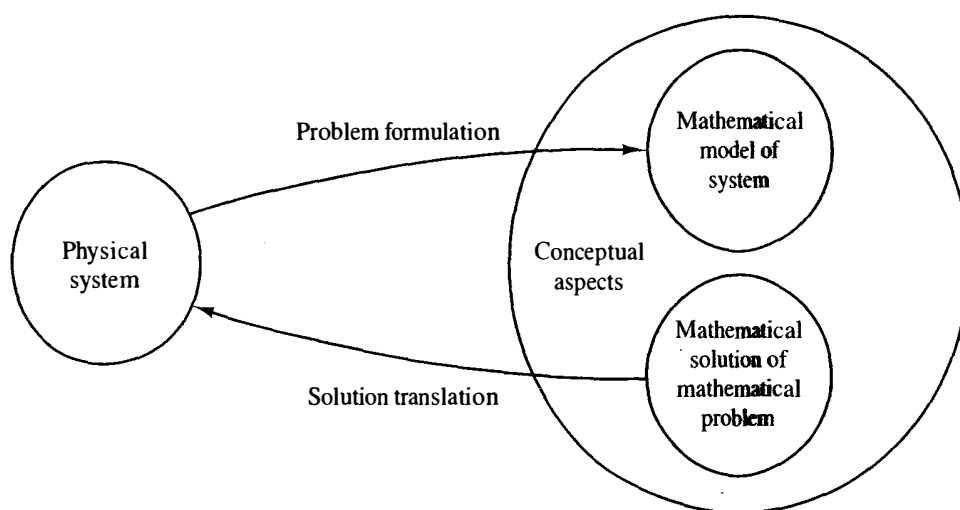


Figure 1-4 Mathematical solutions for physical systems.

the developments in this book. Thus the analytical procedures develop the concepts of control system design and indicate applications of each of the standard controllers.

1.4 SATELLITE MODEL

As the first example of the development of the mathematical model of a physical system, we will consider the attitude control system of a satellite. Assume that the satellite is spherical and has the thruster configuration shown in Figure 1-5. Suppose that $\theta(t)$ is the yaw angle of the satellite. In addition to the thrusters shown, thrusters will also control the pitch angle and the roll angle, giving complete three-axis control of the satellite. We will consider only the yaw-axis control systems, whose purpose is to control the angle $\theta(t)$.

For the satellite, the thrusters, when active, apply a torque $\tau(t)$. The torque of the two active thrusters shown in Figure 1-5 tends to reduce $\theta(t)$. The other two thrusters shown tend to increase $\theta(t)$.

Since there is essentially no friction in the environment of a satellite, and assuming the satellite to be rigid, we can write

$$J \frac{d^2 \theta(t)}{dt^2} = \tau(t) \quad (1-1)$$

where J is the satellite's moment of inertia about the yaw axis. We now derive the transfer function by taking the Laplace transform of (1-1):

$$Js^2 \Theta(s) = T(s) = \mathcal{L}[\tau(t)] \quad (1-2)$$

(Initial conditions are ignored when deriving transfer functions.) Equation (1-2) can be expressed as

$$\frac{\Theta(s)}{T(s)} = G_p(s) = \frac{1}{Js^2} \quad (1-3)$$

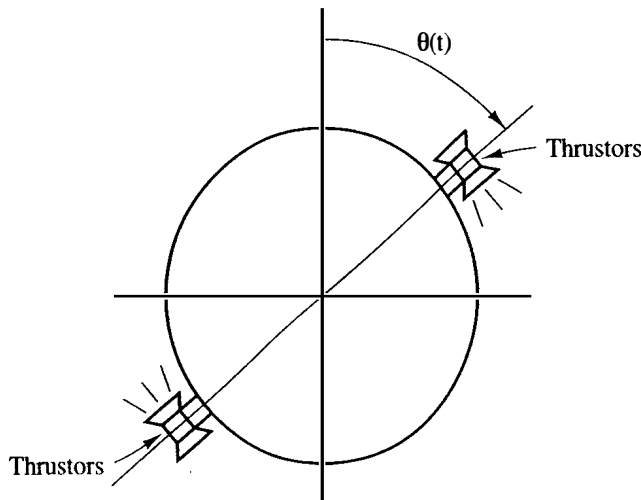


Figure 1-5 Satellite.

The ratio of the Laplace transforms of the output variable $[\theta(t)]$ to input variable $[\tau(t)]$ is called the plant transfer function, and is denoted here as $G_p(s)$. A brief review of the Laplace transform is given in Appendix VII.

The model of the satellite may be specified by either the second-order *differential equation* of (1-1) or the second-order *transfer function* of (1-3). A third model is the state-variable model, which we will now develop. Suppose that we define the variables $x_1(t)$ and $x_2(t)$ as

$$x_1(t) = \theta(t) \quad (1-4)$$

$$x_2(t) = \dot{x}_1(t) = \dot{\theta}(t) \quad (1-5)$$

where $\dot{x}_1(t)$ denotes the derivative of $x_1(t)$ with respect to time. Then, from (1-1) and (1-5),

$$\dot{x}_2(t) = \ddot{\theta}(t) = \frac{1}{J}\tau(t) \quad (1-6)$$

where $\ddot{\theta}(t)$ is the second derivative of $\theta(t)$ with respect to time.

We can now write (1-5) and (1-6) in vector-matrix form (see Appendix IV):

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \tau(t) \quad (1-7)$$

In this equation, $x_1(t)$ and $x_2(t)$ are called the state variables. Hence we may specify the model of the satellite in the form of (1-1), or (1-3), or (1-7). State-variable models of analog systems are considered in greater detail in Chapter 4.

1.5 SERVOMOTOR SYSTEM MODEL

In this section the model of a servo system (a positioning system) is derived. An example of this type of system is an antenna tracking system. In this system, an electric motor is utilized to rotate a radar antenna that tracks an aircraft automatically. The error signal, which is proportional to the difference between the pointing direction of the antenna and the line of sight to the aircraft, is amplified and drives the motor in the appropriate direction so as to reduce this error.

A dc motor system is shown in Figure 1-6. The motor is armature controlled with a constant field. The armature resistance and inductance are R_a and L_a , respectively. We assume that the inductance L_a can be ignored, which is the case for many servomotors. The motor back emf $e_m(t)$ is given by [5]

$$e_m(t) = K_b \omega(t) = K_b \frac{d\theta(t)}{dt} \quad (1-8)$$

where $\theta(t)$ is the shaft position, $\omega(t)$ is the shaft angular velocity, and K_b is a motor-dependent constant. The total moment of inertia connected to the motor shaft

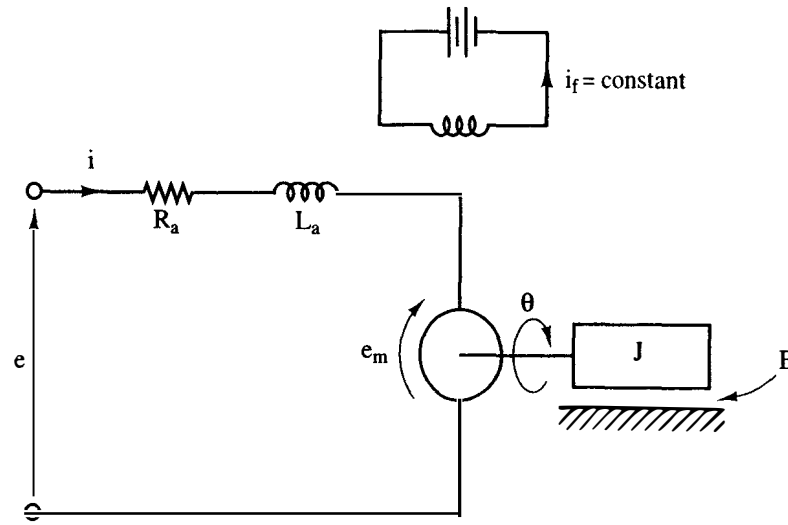


Figure 1-6 Servomotor system.

is J , and B is the total viscous friction. Letting $\tau(t)$ be the torque developed by the motor, we write

$$\tau(t) = J \frac{d^2 \theta(t)}{dt^2} + B \frac{d\theta(t)}{dt} \quad (1-9)$$

The developed torque for this motor is given by

$$\tau(t) = K_T i(t) \quad (1-10)$$

where $i(t)$ is the armature current and K_T is a parameter of the motor. The final equation required is the voltage equation for the armature circuit:

$$e(t) = i(t)R_a + e_m(t) \quad (1-11)$$

These four equations may be solved for the output $\theta(t)$ as a function of the input $e(t)$. First, from (1-11) and (1-8),

$$i(t) = \frac{e(t) - e_m(t)}{R_a} = \frac{e(t)}{R_a} - \frac{K_b}{R_a} \frac{d\theta(t)}{dt} \quad (1-12)$$

Then, from (1-9), (1-10), and (1-12),

$$\tau(t) = K_T i(t) = \frac{K_T}{R_a} e(t) - \frac{K_T K_b}{R_a} \frac{d\theta(t)}{dt} = J \frac{d^2 \theta(t)}{dt^2} + B \frac{d\theta(t)}{dt} \quad (1-13)$$

This equation may be written as

$$J \frac{d^2 \theta(t)}{dt^2} + \frac{BR_a + K_T K_b}{R_a} \frac{d\theta(t)}{dt} = \frac{K_T}{R_a} e(t) \quad (1-14)$$

which is the desired model. This model is second order; if the armature inductance cannot be neglected, the model is third order [6].

Next we take the Laplace transform of (1-14) and solve for the transfer function:

$$\frac{\Theta(s)}{E(s)} = G_p(s) = \frac{K_T/R_a}{Js^2 + \frac{BR_a + K_T K_b}{R_a}s} = \frac{K_T/JR_a}{s\left(s + \frac{BR_a + K_T K_b}{JR_a}\right)} \quad (1-15)$$

Many of the examples of this book are based on this transfer function.

The state-variable model of this system is derived as in the preceding section.

Let

$$\begin{aligned} x_1(t) &= \theta(t) \\ x_2(t) &= \dot{\theta}(t) = \dot{x}_1(t) \end{aligned} \quad (1-16)$$

Then, from (1-14),

$$\dot{x}_2(t) = \ddot{\theta}(t) = -\frac{BR_a + K_T K_b}{JR_a}x_2(t) + \frac{K_T}{JR_a}e(t) \quad (1-17)$$

Hence the state equations may be written as

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{BR_a + K_T K_b}{JR_a} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_T}{JR_a} \end{bmatrix} e(t) \quad (1-18)$$

Antenna Pointing System

We define a *servomechanism*, or more simply, a *servo*, as a system in which mechanical position is controlled. Two servo systems, which in this case form an antenna pointing system, are illustrated in Figure 1-7. The top view of the pedestal illustrates the *yaw-axis* control system. The yaw angle, $\theta(t)$, is controlled by the electric motor and gear system (the control actuator) shown in the side view of the pedestal.

The pitch angle, $\phi(t)$, is shown in the side view. This angle is controlled by a motor and gear system within the pedestal; this actuator is not shown.

We consider only the yaw-axis control system. The electric motor rotates the antenna and the sensor, which is a digital shaft encoder [7]. The output of the encoder is a binary number that is proportional to the angle of the shaft. For this example, a digital-to-analog converter (discussed in Chapter 3) is used to convert the binary number to a voltage $v_o(t)$ that is proportional to the angle of rotation of the shaft. Later we consider examples in which the binary number is transmitted directly to a digital controller.

In Figure 1-7a the voltage $v_o(t)$ is directly proportional to the yaw angle of the antenna, and the voltage $v_i(t)$ is directly proportional (same proportionality constant) to the desired yaw angle. If the yaw angle and the desired yaw angle are different, the error voltage $e(t)$ is nonzero. This voltage is amplified and applied to the motor to cause rotation of the motor shaft in the direction that reduces the error voltage.

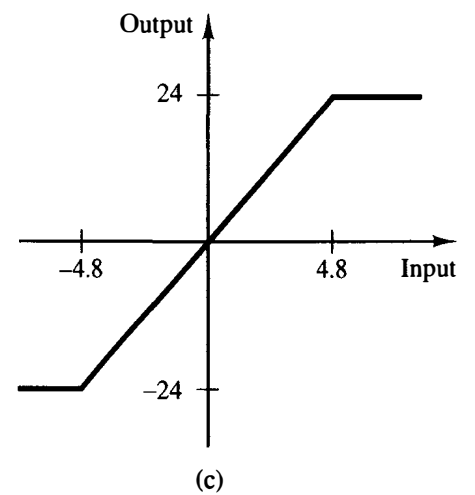
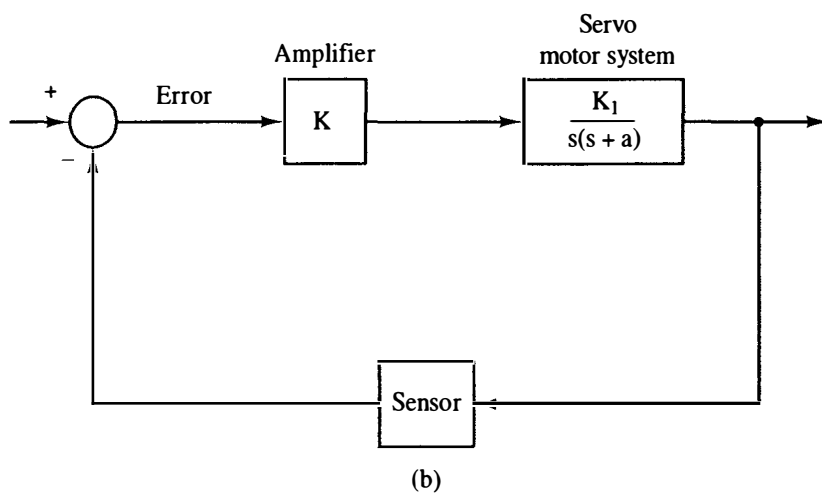
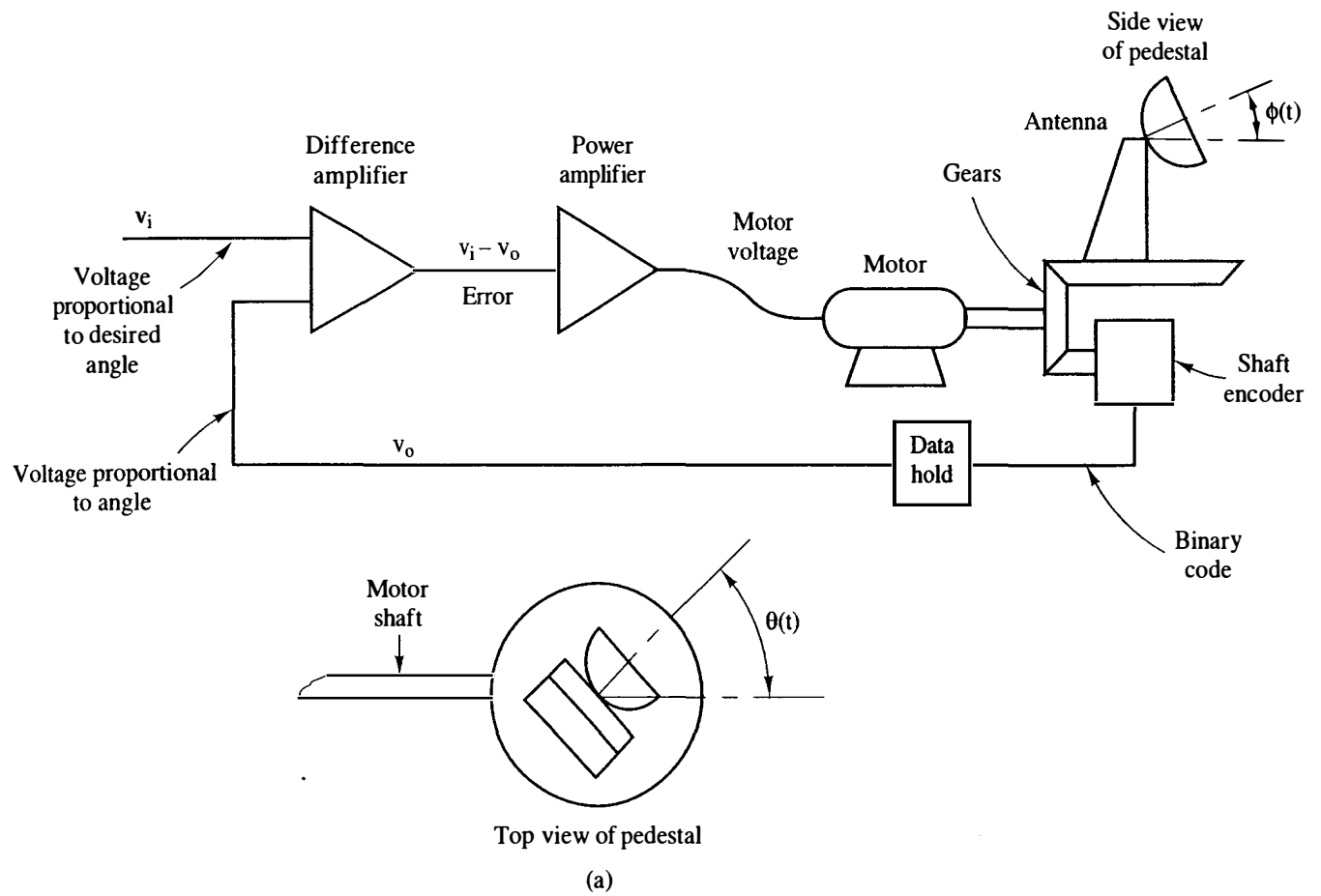


Figure 1-7 Servo control system.

The system *block diagram* is given in Figure 1-7b. Since the error signal is normally a low-power signal, a power amplifier is required. However, this amplifier introduces a nonlinearity in the system. The output voltage and can be saturated at a maximum value. Suppose that the amplifier has a gain of 5 and a maximum output of 24 V. The amplifier input-output characteristic is as shown in Figure 1-7c. The system is nonlinear for an error signal larger than 4.8 V.

In many control systems, we go to great lengths to ensure that the system operation is confined to linear regions and avoid nonlinear operation. For example, in a motor control system, we must apply maximum voltage to the motor to achieve maximum response. Thus for large error signals we would have the amplifier saturate.

The analysis and design of nonlinear systems is beyond the scope of this book; we will always assume that the system is operating in a linear mode.

Robotic Control System

A line drawing of an industrial robot is shown in Figure 1-8. The basic element of the control system for each joint of the robot is a servomotor. We take the usual

assumption that the error signal is required to drive the motor. Since the error signal is normally a low-power signal, a power amplifier is required. However, this amplifier introduces a nonlinearity in the system. The output voltage and can be saturated at a maximum value. Suppose that the amplifier has a gain of 5 and a maximum output of 24 V. The amplifier input-output characteristic is as shown in Figure 1-7c. The system is nonlinear for an error signal larger than 4.8 V.

In many control systems, we go to great lengths to ensure that the system operation is confined to linear regions and avoid nonlinear operation. For example, in a motor control system, we must apply maximum voltage to the motor to achieve maximum response. Thus for large error signals we would have the amplifier saturate.

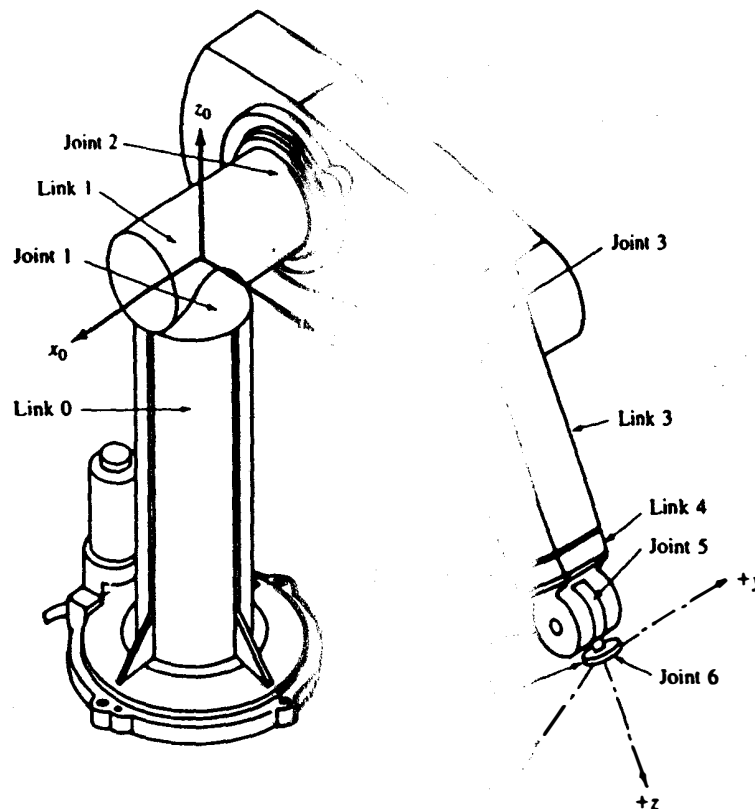


Figure 1-8 Industrial robot. (Courtesy of the University of Michigan and Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, John Wiley & Sons, Inc., 1987.)

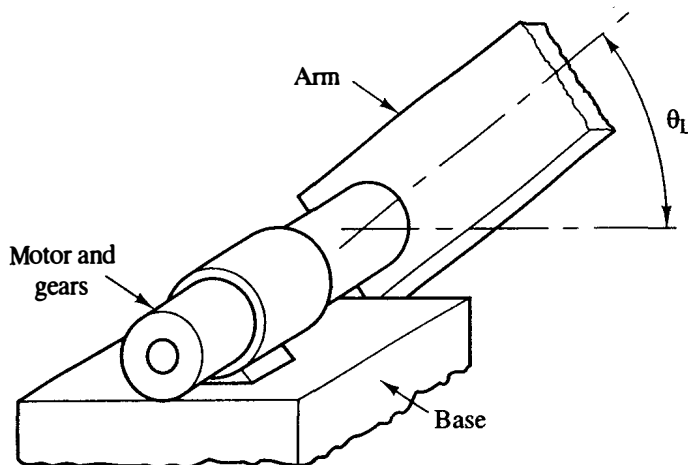
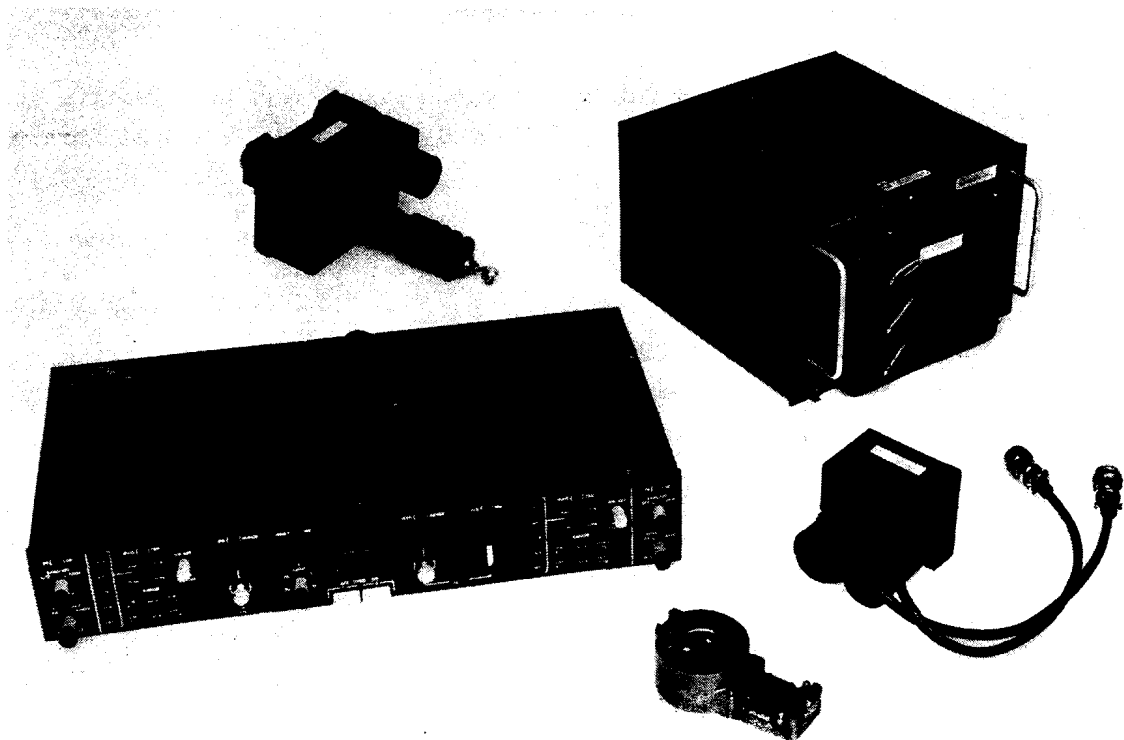


Figure 1-9 Robot arm joint. (From Phillips and Harbor, *Feedback Control Systems*, 2d ed., Prentice Hall, 1991, Fig. 2.43.)

approach of considering each joint of the robot as a simple servomechanism, and ignore the movements of the other joints in the arm. Although this approach is simple in terms of analysis and design, the result is often a less than desirable control of the joint [8].

Figure 1-9 illustrates the single joint of a robot arm. The actuator is assumed to be a servomotor of the type just described. In addition, it is assumed that the arm is attached to the motor through gears, with a gear ratio of n [8].



Hardware for an automated flight control system, clockwise from the lower left corner; the glareshield control panel, the elevator load feel/flap limiter, the flight control computer, the autothrottle servo, and the control wheel sensor. (Courtesy of Honeywell Inc., Sperry Commercial FLight Systems Group.)

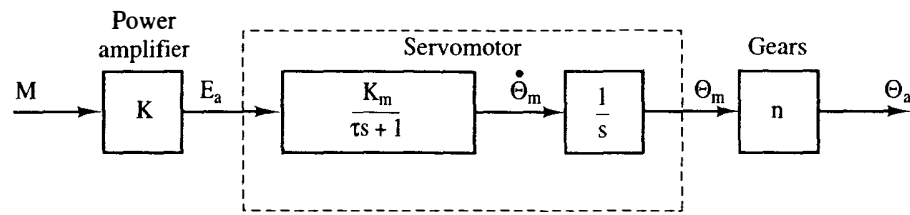


Figure 1-10 Model of robot arm joint.

The model of the robot arm joint is given in Figure 1-10, where the second-order model of the servomotor is assumed. If the armature inductance of the motor cannot be ignored, the model is third order [8]. In this model, $E_a(s)$ is the armature voltage, and is used to control the position of the arm. The input signal $M(s)$ is assumed to be from a digital computer, and the power amplifier is required since a computer output signal cannot drive the motor. The angle of the motor shaft is $\Theta_m(s)$, and the angle of the arm is $\Theta_a(s)$. As described above, the inertia and friction of both the gears and the arm are included in the servomotor model, and hence the model shown is the complete model of the robot joint. This model will be used in several problems that appear at the ends of the chapters.

1.6 TEMPERATURE CONTROL SYSTEM

As a third example of modeling, a thermal system will be considered. It is desired to control the temperature of a liquid in a tank. Liquid is flowing out at some rate, being replaced by liquid at temperature $\tau_i(t)$ as shown in Figure 1-11. A mixer

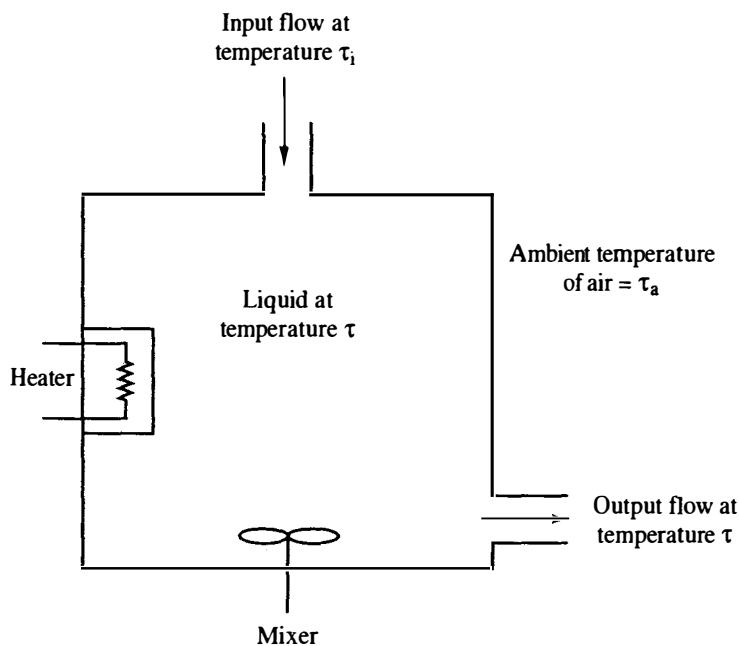


Figure 1-11 Thermal system.

agitates the liquid such that the liquid temperature can be assumed uniform at a value $\tau(t)$ throughout the tank. The liquid is heated by an electric heater.

We first make the following definitions:

$q_e(t)$ = heat flow supplied by the electric heater

$q_i(t)$ = heat flow via liquid entering the tank

$q_l(t)$ = heat flow into the liquid

$q_o(t)$ = heat flow via liquid leaving the tank

$q_s(t)$ = heat flow through the tank surface

By the conservation of energy, heat added to the tank must equal that stored in the tank plus that lost from the tank. Thus

$$q_e(t) + q_i(t) = q_l(t) + q_o(t) + q_s(t) \quad (1-19)$$

Now [9]

$$q_l(t) = C \frac{d\tau(t)}{dt} \quad (1-20)$$

where C is the thermal capacity of the liquid in the tank. Letting $v(t)$ equal the flow into and out of the tank (assumed equal) and H equal the specific heat of the liquid, we can write

$$q_i(t) = v(t)H\tau_i(t) \quad (1-21)$$

and

$$q_o(t) = v(t)H\tau(t) \quad (1-22)$$

Let $\tau_a(t)$ be the ambient temperature outside the tank and R be the thermal resistance to heat flow through the tank surface. Then

$$q_s(t) = \frac{\tau(t) - \tau_a(t)}{R} \quad (1-23)$$

Substituting (1-20) through (1-23) into (1-19) yields

$$q_e(t) + v(t)H\tau_i(t) = C \frac{d\tau(t)}{dt} + v(t)H\tau(t) + \frac{\tau(t) - \tau_a(t)}{R}$$

We now make the assumption that the flow $v(t)$ is constant with the value V ; otherwise, the last differential equation is time-varying. Then

$$q_e(t) + VH\tau_i(t) = C \frac{d\tau(t)}{dt} + VH\tau(t) + \frac{\tau(t) - \tau_a(t)}{R} \quad (1-24)$$

This model is a first-order linear differential equation with constant coefficients. In terms of a control system, $q_e(t)$ is the control input signal, $\tau_i(t)$ and $\tau_a(t)$ are disturbance input signals, and $\tau(t)$ is the output signal.

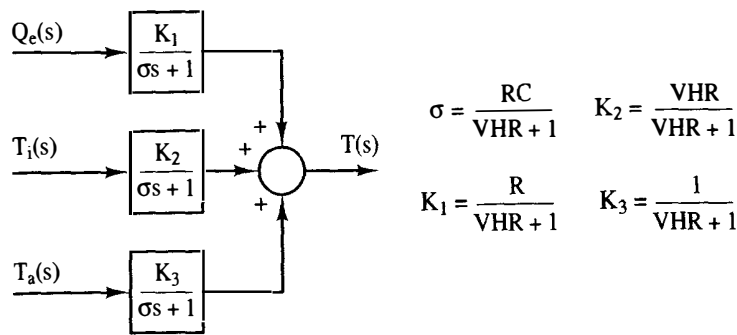


Figure 1-12 Block diagram of a thermal system.

Taking the Laplace of (1-24) and solving for $T(s) = \mathcal{L}[\tau(t)]$ yields

$$T(s) = \frac{Q_e(s)}{Cs + VH + (1/R)} + \frac{VHT_i(s)}{Cs + VH + (1/R)} + \frac{(1/R)T_a(s)}{Cs + VH + (1/R)} \quad (1-25)$$

Different configurations may be used to express (1-25) as a block diagram; one is given in Figure 1-12.

If we ignore the disturbance inputs, the transfer-function model of the system is simple and first order. However, at some step in the control system design the disturbances must be considered. Quite often a major specification in a control system design is the minimization of system response to disturbance inputs.

The model developed in this section also applies directly to the control of the air temperature in an oven or a test chamber. For many of these systems, no air is introduced from the outside; hence the disturbance input $q_i(t)$ is zero. Of course, the parameters for the liquid in (1-25) are replaced with those for air.

1.7 SUMMARY

In this chapter we have introduced the concepts of a closed-loop control system. Next, models of three physical systems were discussed. First, a model of a satellite was derived. Next, the model of a servomotor was developed; then two examples, an antenna pointing system and a robot arm, were discussed. Finally, a model was developed for control of the temperature of a tank of liquid. These systems are continuous time, and generally, the Laplace transform is used in the analysis and design of these systems. In the next chapter we extend the concepts of this chapter to a system controlled by a digital computer and introduce some of the mathematics required to analyze and design this type of system.

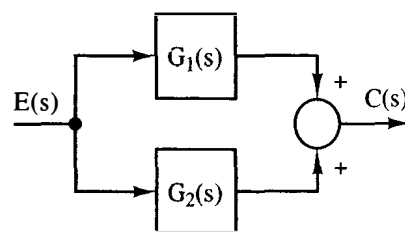
REFERENCES

1. M. Athans, M. L. Dertouzos, R. N. Spann, and S. J. Mason, *Systems, Networks, and Computations: Multivariable Methods*. New York: McGraw-Hill Book Company, 1974.

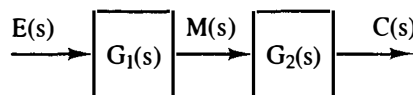
2. R. F. Wigginton, "Evaluation of OPS-II Operational Program for the Automatic Carrier Landing System," Naval Electronic Systems Test and Evaluation Facility, Saint Inigoes, MD, 1971.
3. C. L. Phillips, E. R. Graf, and H. T. Nagle, Jr., "MATCALS Error and Stability Analysis," Report AU-EE-75-2080-1, Auburn University, Auburn, AL, 1975.
4. W. A. Gardner, *Introduction to Random Processes*. New York: Macmillan Publishing Company, 1986.
5. A. E. Fitzgerald, C. Kingsley, and S. D. Umans, *Electric Machinery*, 5th ed. New York: McGraw-Hill Book Company, 1990.
6. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2d ed. Englewood Cliffs, NJ: Prentice Hall, 1991.
7. C. W. deSilva, *Control Sensors and Actuators*. Englewood Cliffs, NJ: Prentice Hall, 1989.
8. K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill Book Company, 1987.
9. J. D. Trimmer, *Response of Physical Systems*. New York: John Wiley & Sons, Inc., 1950.

PROBLEMS

- 1-1. (a) Show that the transfer function of two systems in parallel, as shown in Figure P1-1a, is equal to the sum of the transfer functions.
- (b) Show that the transfer function of two systems in series (cascade), as shown in Figure P1-1b, is equal to the product of the transfer functions.



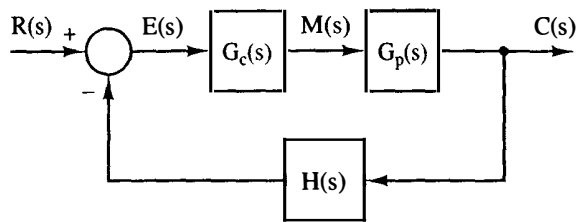
(a)



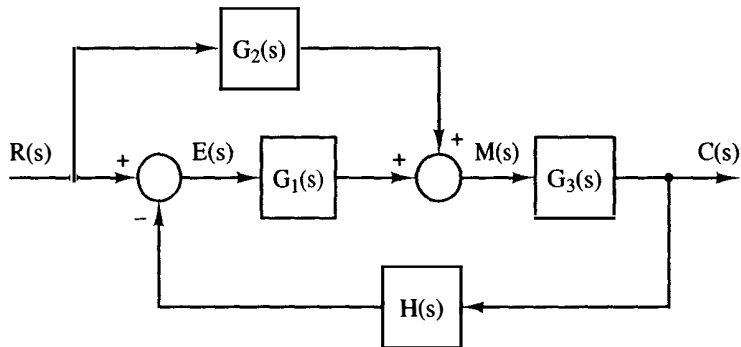
(b)

Figure P1-1 Systems for Problem 1-1.

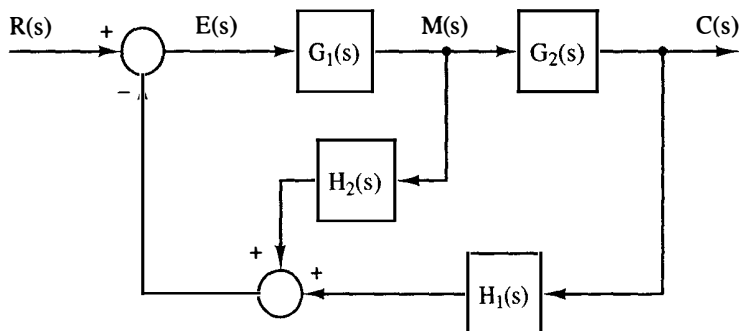
- 1-2. By writing algebraic equations and eliminating variables, calculate the transfer function $C(s)/R(s)$ for the system of:
- (a) Figure P1-2a.
 - (b) Figure P1-2b.
 - (c) Figure P1-2c.



(a)



(b)



(c)

Figure P1-2 Systems for Problem 1-2.

1-3. Use Mason's gain formula of Appendix II to verify the results of Problem P1-2 for the system of:

- (a) Figure P1-2a.
- (b) Figure P1-2b.
- (c) Figure P1-2c.

1-4. A feedback control system is illustrated in Figure P1-4. The plant transfer function is given by

$$G_p(s) = \frac{5}{0.2s + 1}$$

- (a) Write the differential equation of the plant. This equation relates $c(t)$ and $m(t)$.
- (b) Modify the equation of part (a) to yield the system differential equation; this

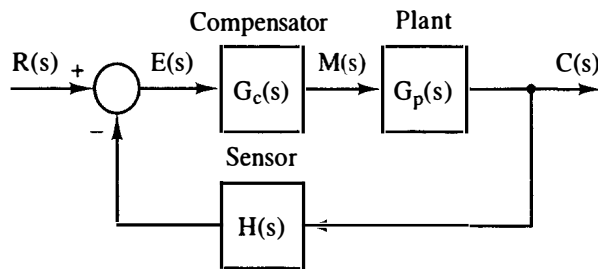


Figure P1-4 Feedback control system.

equation relates $c(t)$ and $r(t)$. The compensator and sensor transfer functions are given by

$$G_c(s) = 10, \quad H(s) = 1$$

- (c) Derive the system transfer function from the results of part (b).
 (d) It is shown in Problem 1-2(a) that the closed-loop transfer function of the system of Figure P1-4 is given by

$$\frac{C(s)}{R(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)H(s)}$$

Use this relationship to verify the results of part (c).

- (e) Recall that the transfer-function pole term $(s + a)$ yields a time constant $\tau = 1/a$, where a is real. Find the time constants for both the open-loop and closed-loop systems.

1-5. Repeat Problem 1-4 with the transfer functions

$$G_c(s) = 2, \quad G_p(s) = \frac{3s + 8}{s^2 + 2s + 2}, \quad H(s) = 1$$

For part (e), recall that the transfer-function underdamped pole term $[(s + a)^2 + b^2]$ yields a time constant $\tau = 1/a$.

1-6. Repeat Problem 1-4 with the transfer functions

$$G_c(s) = 2, \quad G_p(s) = \frac{5}{s^2 + 2s + 2}, \quad H(s) = 3s + 1$$

- 1-7.** The antenna positioning system described in Section 1.5 is shown in Figure P1-7. In this problem we consider the yaw angle control system, where $\theta(t)$ is the yaw angle. Suppose that the gain of the power amplifier is 10 V/V, and that the gear ratio and the angle sensor (the shaft encoder and the data hold) are such that

$$v_o(t) = 0.04\theta(t)$$

where the units of $v_o(t)$ are volts and of $\theta(t)$ are degrees. Let $e(t)$ be the input voltage to the motor; the transfer function of the motor pedestal is given as

$$\frac{\Theta(s)}{E(s)} = \frac{20}{s(s + 6)}$$

- (a) With the system open loop [$v_o(t)$ is always zero], a unit-step function of voltage is applied to the motor [$E(s) = 1/s$]. Consider only the *steady-state response*. Find the

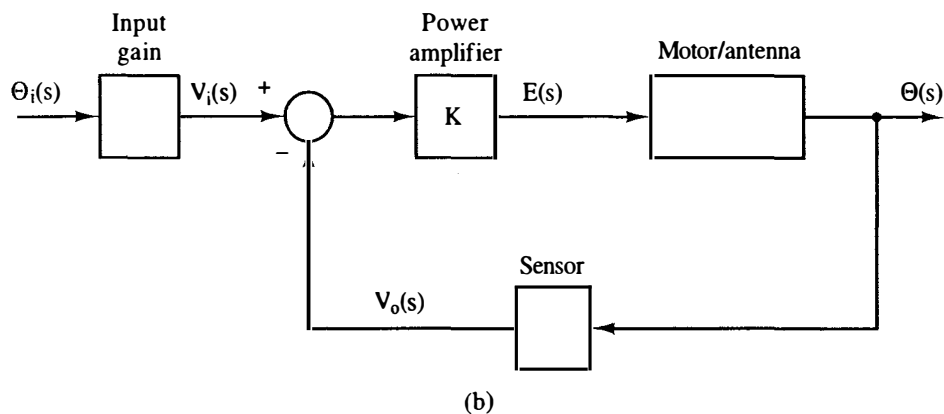
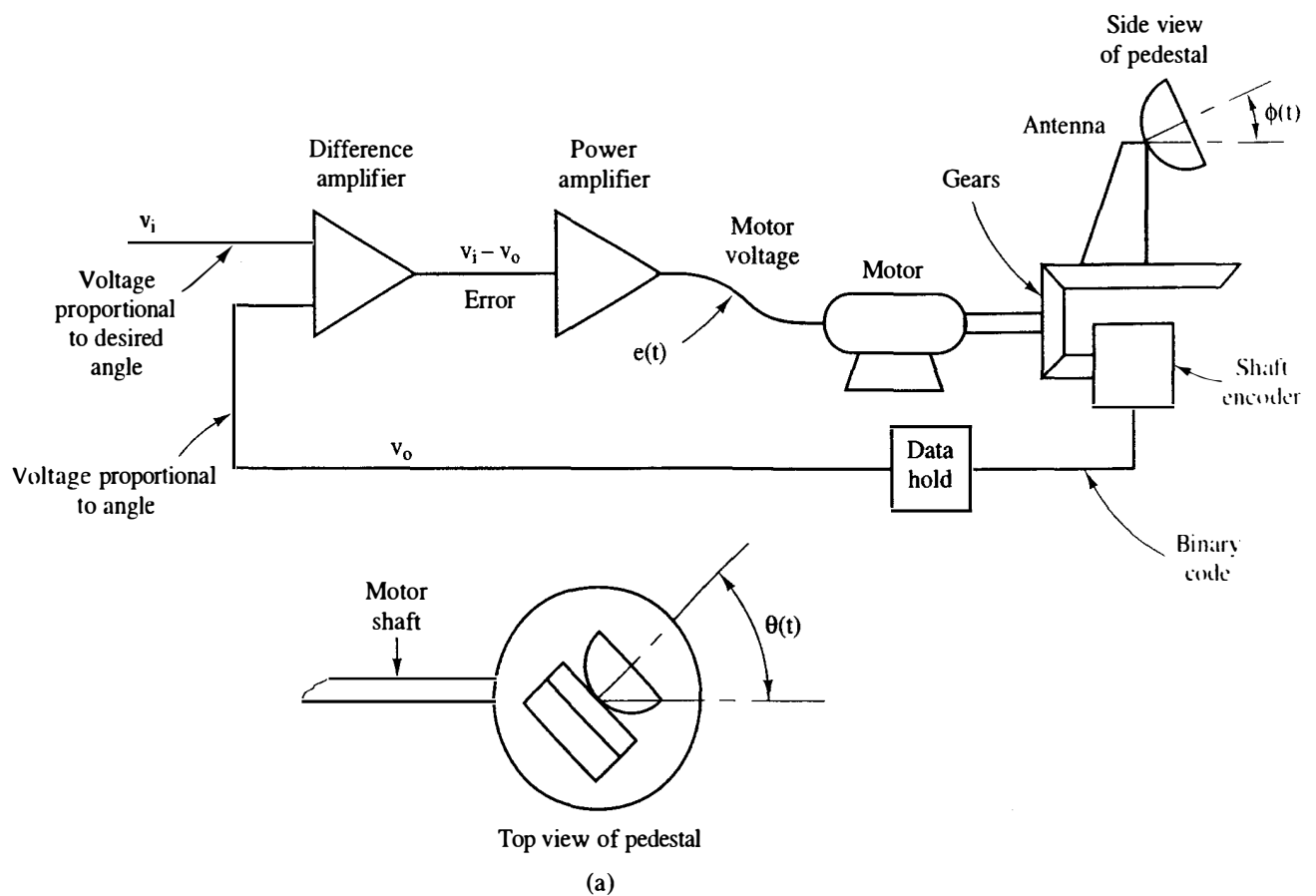


Figure P1-7 System for Problem 1-7.

output angle $\theta(t)$ in degrees, and the angular velocity of the antenna pedestal, $\dot{\theta}(t)$, in both degrees per second and rpm.

- (b) The system block diagram is given in Figure P1-7b, with the angle signals shown in degrees and the voltages in volts. Add the required gains and the transfer functions to this block diagram.
- (c) Make the changes necessary in the gains in part (b) such that the units of $\theta(t)$ are radians.

- (d) A step input of $\theta_i(t) = 10^\circ$ is applied at the system input at $t = 0$. Find the response $\theta(t)$.
- (e) The response in part (d) reaches steady state in approximately how many seconds?
- 1-8.** The state-variable model of a servomotor is given in Section 1.5. Expand these state equations to model the antenna pointing system of Problem 1-7(b).
- 1-9.** (a) Find the transfer function $\Theta(s)/\Theta_i(s)$ for the antenna pointing system of Problem 1-7(b). This transfer function yields the angle $\theta(t)$ in degrees.
- (b) Modify the transfer function in part (a) such that use of the modified transfer function yields $\theta(t)$ in radians.
- (c) Verify the results of part (b) using the block diagram of Problem 1-7(b).
- 1-10.** A thermal test chamber is illustrated in Figure P1-10a. This chamber, which is a large room, is used to test large devices under various thermal stresses. The chamber is heated with steam, which is controlled by an electrically activated valve. The temperature of the chamber is measured by a sensor based on a thermistor, which is a semiconductor resistor whose resistance varies with temperature. Opening the door into the chamber affects the chamber temperature and thus must be considered as a disturbance.

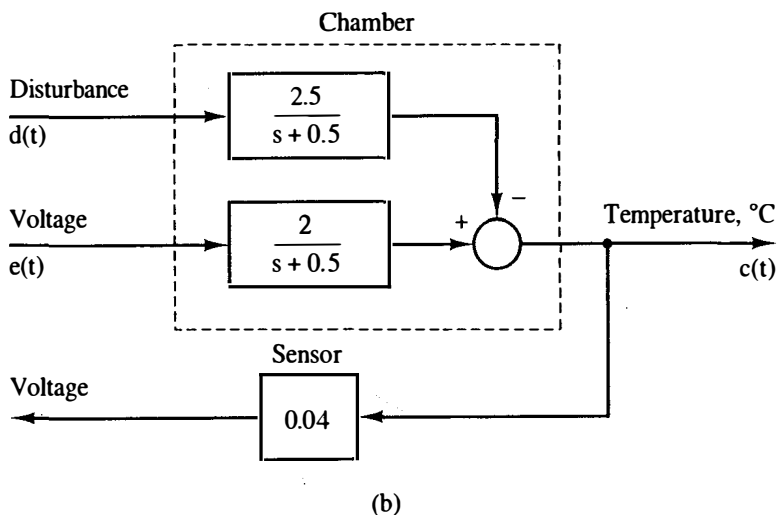
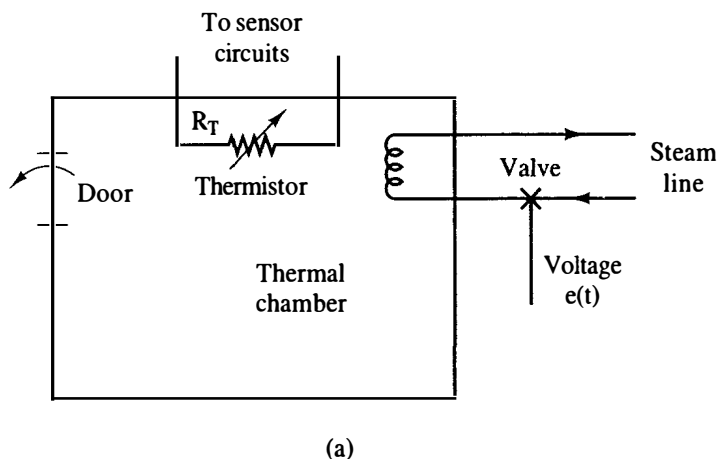


Figure P1-10 A thermal stress chamber.

A simplified model of the test chamber is shown in Figure P1-10b, with the units of time in minutes. The control input is the voltage $e(t)$, which controls the valve in the steam line, as shown. For the disturbance $d(t)$, a unit step function is used to model the opening of the door. With the door closed, $d(t) = 0$.

- (a) Find the time constant of the chamber.
 - (b) With the controlling voltage $e(t) = 5u(t)$ and the chamber door closed, find and plot the chamber temperature $c(t)$. In addition, give the steady-state temperature.
 - (c) A tacit assumption in part (a) is an initial chamber temperature of zero degrees Celsius. Repeat part (b), assuming that the initial chamber temperature is $c(0) = 25^\circ\text{C}$.
 - (d) Two minutes after the application of the voltage in part (c), the door is opened, and it remains open. Add the effects of this disturbance to the plot of part (c).
 - (e) The door in part (d) remains open for 12 min. and is then closed. Add the effects of this disturbance to the plot of part (d).
- 1-11.** The thermal chamber transfer function $C(s)/E(s) = 2/(s + 0.5)$ of Problem 1-10 is based on the units of time being minutes.
- (a) Modify this transfer function to yield the chamber temperature $c(t)$ based on seconds.
 - (b) Verify the result in part (a) by solving for $c(t)$ with the door closed and the input $e(t) = 5u(t)$ volts, (i) using the chamber transfer function found in part (a), and (ii) using the transfer function of Figure P1-10. Show that (i) and (ii) yield the same temperature at $t = 1$ min.
- 1-12.** The satellite of Section 1.4 is connected in the closed-loop control system shown in Figure P1-12. The torque is directly proportional to the error signal.
- (a) Derive the transfer function $\Theta(s)/\Theta_c(s)$, where $\theta(t) = \mathcal{L}^{-1}[\Theta(s)]$ is the commanded attitude angle.
 - (b) The state equations for the satellite are derived in Section 1.4. Modify these equations to model the closed-loop system of Figure P1-12.
- 1-13.** (a) In the system of Problem 1-12, $J = 0.4$ and $K = 14.4$, in appropriate units. The attitude of the satellite is initially at 0° . At $t = 0$, the attitude is commanded to 20° ; that is, a 20° step is applied at $t = 0$. Find the response $\theta(t)$.
- (b) Repeat part (a), with the initial conditions $\theta(0) = 10^\circ$ and $\dot{\theta}(0) = 30^\circ/\text{s}$. Note that we have assumed that the units of time for the system is seconds.
 - (c) Verify the solution in part (b) by first checking the initial conditions and then substituting the solution into the system differential equation.
- 1-14.** The input to the satellite system of Figure P1-12 is a step function $\theta_c(t) = 5u(t)$ in degrees. As a result, the satellite angle $\theta(t)$ varies sinusoidally at a frequency of 10 cycles per minute. Find the amplifier gain K and the moment of inertia J for the system, assuming that the units of time in the system differential equation are seconds.
- 1-15.** The satellite control system of Figure P1-12 is not usable, since the response to any excitation includes an undamped sinusoid. The usual compensation for this system involves measuring the angular velocity $d\theta(t)/dt$. The feedback signal is then a linear sum of the position signal $\theta(t)$ and the velocity signal $d\theta(t)/dt$. This system is depicted in Figure P1-15, and is said to have *rate feedback*.
- (a) Derive the transfer function $\Theta(s)/\Theta_c(s)$ for this system.
 - (b) The state equations for the satellite are derived in Section 1.4. Modify these equations to model the closed-loop system of Figure P1-15.

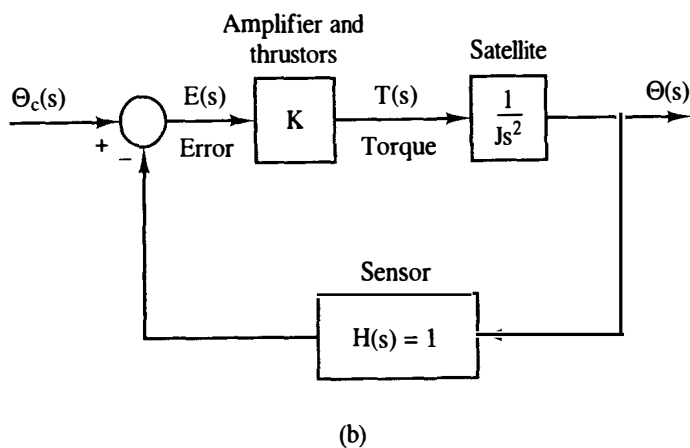
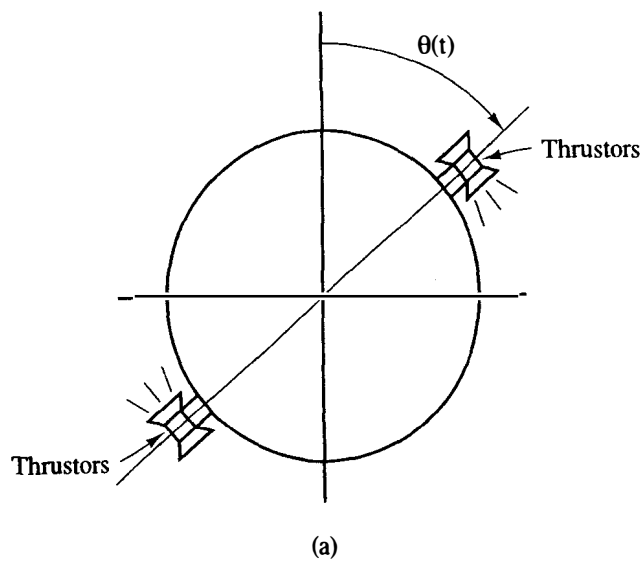


Figure P1-12 Satellite control system.

(c) The state equations in part (b) can be expressed as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\theta_c(t)$$

The system characteristic equation is

$$|s\mathbf{I} - \mathbf{A}| = 0$$

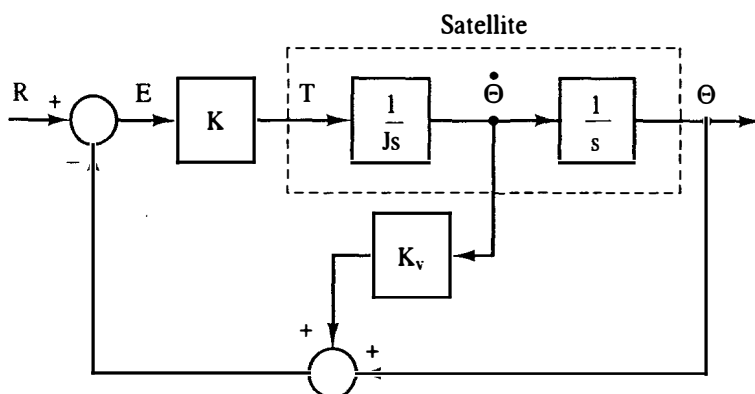


Figure P1-15 Satellite control system with rate feedback.

Show that $|sI - A|$ in part (b) is equal to the transfer function denominator in part (a).

- 1-16. Shown in Figure P1-16 is the block diagram of one joint of a robot arm. This system is described in Section 1.5. The input $M(s)$ is the controlling signal, $E_a(s)$ is the servomotor input voltage, $\Theta_m(s)$ is the motor shaft angle, and the output $\Theta_a(s)$ is the angle of the arm. The inductance of the armature of the servomotor has been neglected such that the servomotor transfer function is second order. The servomotor transfer function includes the inertia of both the gears and the robot arm. Derive the transfer functions $\Theta_a(s)/M(s)$ and $\Theta_a(s)/E_a(s)$.

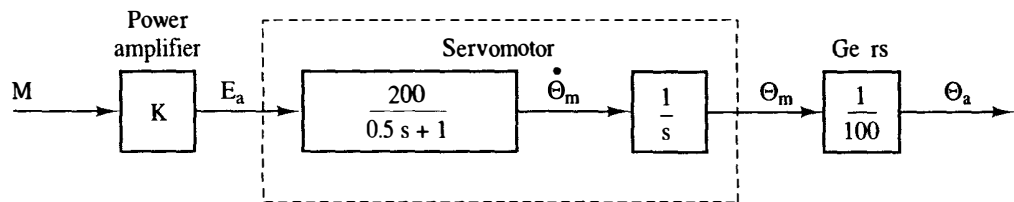


Figure P1-16 A model of a robot arm.

- 1-17. Consider the robot arm depicted in Figure P1-16.
- Suppose that the units of $e_a(t)$ are volts, that the units of both $\theta_m(t)$ and $\theta_a(t)$ are degrees, and that the units of time is seconds. If the servomotor is rated at 24 V [the voltage $e_a(t)$ should be less than or equal to 24 V], find the rated rpm of the motor (the motor rpm, in steady state, with 24 V applied).
 - Find the maximum rate of movement of the robot arm, in degrees per second, with a step voltage of $e_a(t) = 24u(t)$ volts applied.
 - Assume that $e_a(t)$ is a step function of 24 V. Give the time required for the arm to be moving at 99 percent of the maximum rate of movement found in part (b).
 - Suppose that the input $m(t)$ is constrained by system hardware to be less than or equal to 10 V in magnitude. What value would you choose for the gain K . Why?

Discrete-Time Systems and the z-Transform

2.1 INTRODUCTION

In this chapter two important topics are introduced: discrete-time systems and the z-transform. In contrast to a continuous-time system whose operation is described (modeled) by a set of differential equations, a discrete-time system is one whose operation is described by a set of difference equations. The transform method employed in the analysis of linear time-invariant continuous-time systems is the Laplace transform; in a similar manner, the transform used in the analysis of linear time-invariant discrete-time systems is the z-transform. The modeling of discrete-time systems by difference equations, transfer functions, and state equations is presented in this chapter.

2.2 DISCRETE-TIME SYSTEMS

To illustrate the idea of a discrete-time system, consider the digital control system shown in Figure 2-1a. The digital computer performs the compensation function within the system. The interface at the input of the computer is an analog-to-digital (A/D) converter, and is required to convert the error signal, which is a continuous-time signal, into a form that can be readily processed by the computer. At the computer output a digital-to-analog (D/A) converter is required to convert the binary signals of the computer into a form necessary to drive the plant.

We will now consider the following example. Suppose that the A/D converter, the digital computer, and the D/A converter are to replace an analog, or continuous-

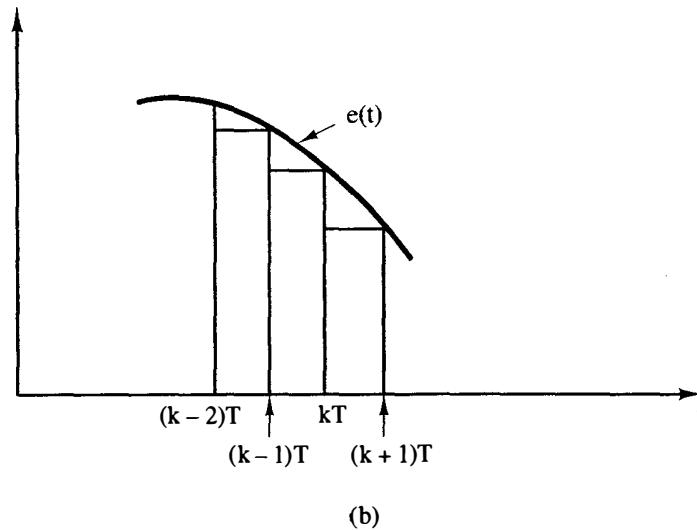
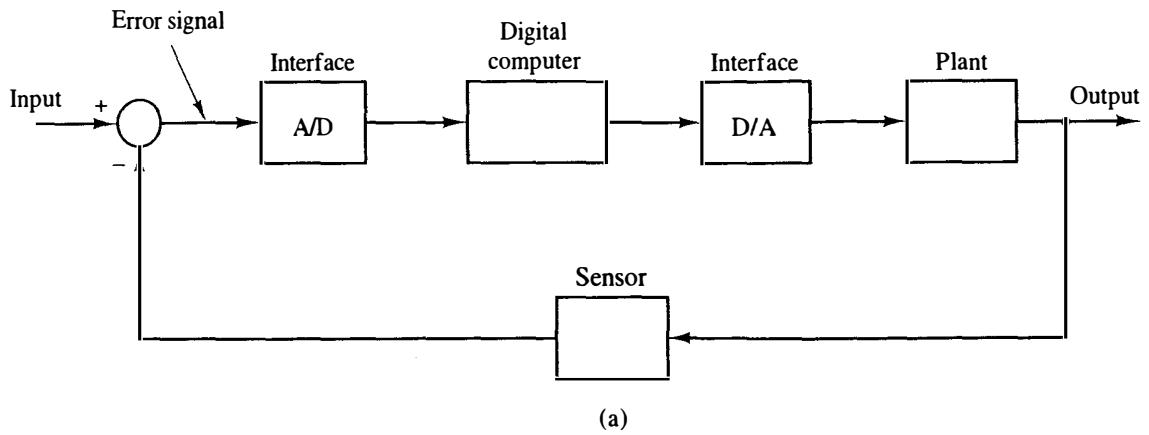


Figure 2-1 Digital control system.

time, proportional-integral (PI) compensator such that the digital control system response has essentially the same characteristics as the analog system. (The PI controller is discussed in Chapter 8.) The analog controller output is given by

$$m(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau \quad (2-1)$$

where $e(t)$ is the controller input signal, $m(t)$ is the controller output signal, and K_P and K_I are constant gains determined by the design process.

Since the digital computer can be programmed to multiply, add, and integrate numerically, the controller equation can be realized using the digital computer. For this example, the rectangular rule of numerical integration [1], illustrated in Figure 2-1b, will be employed. Of course, other algorithms of numerical integration may also be used.

For the rectangular rule, the area under the curve in Figure 2-1b is approximated by the sum of the rectangular areas shown. Thus, letting $x(t)$ be the numerical integral of $e(t)$, we can write

$$x(kT) = x[(k-1)T] + Te(kT) \quad (2-2)$$

where T is the numerical algorithm step size, in seconds. Then (2-1) becomes, for the digital compensator,

$$m(kT) = K_P e(kT) + K_I x(kT)$$

Equation (2-2) is a first-order linear *difference equation*. The general form of a first-order linear time-invariant difference equation is (with the T omitted for convenience)

$$x(k) = b_1 e(k) + b_0 e(k-1) - a_0 x(k-1) \quad (2-3)$$

This equation is first order since the signals from only the last sampling instant appear explicitly in the equation. The general form of an n th-order linear difference equation is

$$\begin{aligned} x(k) = & b_n e(k) + b_{n-1} e(k-1) + \cdots + b_0 e(k-n) \\ & - a_{n-1} x(k-1) - \cdots - a_0 x(k-n) \end{aligned} \quad (2-4)$$

It will be shown in Chapter 5 that if the plant in Figure 2-1 is also linear and time invariant, the entire system may be modeled by a difference equation of the form of (2-4), which is generally of higher order than that of the controller. Compare (2-4) to a linear differential equation describing an n th-order continuous-time system.

$$\begin{aligned} y(t) = & \beta_n \frac{d^n e(t)}{dt^n} + \cdots + \beta_1 \frac{de(t)}{dt} + \beta_0 e(t) \\ & - \alpha_n \frac{d^n y(t)}{dt^n} - \cdots - \alpha_1 \frac{dy(t)}{dt} \end{aligned} \quad (2-5)$$

Two approaches may be used in the design of digital compensators. First, an analog compensator may be designed and then converted by some approximate procedure to a digital compensator, as in the example above. Several techniques for converting analog compensators to digital compensators are presented in Chapter 11, and the interested reader may cover that chapter after studying the material through Section 2.7, if desired. Chapters 3 through 10 present exact methods of designing digital compensators, as compared to the approximate methods of converting analog compensators to digital compensators.

The describing equation of a linear, time-invariant analog (i.e., continuous-time) filter is also of the form of (2-5). The device that realizes this filter, usually an RC network with operational amplifiers, can be considered to be an analog computer programmed to solve the filter equation. In a like manner, (2-4) is the describing equation of a linear, time-invariant discrete filter, which is usually called a *digital filter*. The device that realizes this filter is a digital computer programmed

to solve (2-4), or a special-purpose computer built specifically to solve (2-4). Thus the digital computer in Figure 2-1 would be programmed to solve a difference equation of the form of (2-4), and the problem of the control system designer would be to determine (1) T , the sampling period; (2) n , the order of the difference equation; and (3) a_i and b_i , the filter coefficients, such that the control system has certain desired characteristics.

There are additional problems in the realization of the digital filter: for example, the computer wordlength required to keep system errors caused by round-off in the computer at an acceptable level. As an example, a digital filter (controller) has been designed and implemented to land aircraft automatically on U.S. Navy aircraft carriers [2]. In this system, the sample rate is 25 Hz ($T = 0.04$ s), and the controller is eleventh order. The minimum word length required for the computer was found to be 32 bits, in order that system errors caused by round-off in the computer remain at acceptable levels. As an additional point, this controller is a proportional-plus-integral-plus-derivative (PID) controller with extensive noise filtering required principally because of the differentiation in the D part of the filter. The integration and the differentiation are performed numerically, as is discussed in Chapters 8 and 11. In many applications other than control systems, digital filters have been designed to replace analog filters and the problems encountered are the same as those listed above.

2.3 TRANSFORM METHODS

In linear time-invariant continuous-time systems, the Laplace transform can be utilized in system analysis and design. For example, an alternative, but equally valid description of the operation of a system described by (2-5) is obtained by taking the Laplace transform of this equation and solving for the transfer function:

$$\frac{Y(s)}{E(s)} = \frac{\beta_n s^n + \cdots + \beta_1 s + \beta_0}{\alpha_n s^n + \cdots + \alpha_1 s + 1} \quad (2-6)$$

A transform will now be defined that can be utilized in the analysis of discrete-time systems modeled by difference equations of the form given in (2-4).

A transform is defined for number sequences as follows. The function $E(z)$ is defined as a power series in z^{-k} with coefficients equal to the values of the number sequence $\{e(k)\}$. This transform, called the z-transform, is then expressed by the transform pair

$$\begin{aligned} E(z) &= \mathcal{Z}[\{e(k)\}] = e(0) + e(1)z^{-1} + e(2)z^{-2} + \cdots \\ e(k) &= \mathcal{Z}^{-1}[E(z)] = \frac{1}{2\pi j} \oint_{\Gamma} E(z) z^{k-1} dz, \quad j = \sqrt{-1} \end{aligned} \quad (2-7)$$

where $\mathcal{Z}(\cdot)$ indicates the z-transform operation and $\mathcal{Z}^{-1}(\cdot)$ indicates the inverse z-transform. $E(z)$ in (2-7) can be written in more compact notation as

$$E(z) = \mathcal{Z}[\{e(k)\}] = \sum_{k=0}^{\infty} e(k)z^{-k} \quad (2-8)$$

For convenience, we often omit the braces and express $\mathcal{Z}[\{e(k)\}]$ as $\mathcal{Z}[e(k)]$. However, it should be remembered that the z -transform applies to a sequence.

The z -transform is defined for any number sequence $\{e(k)\}$, and may be used in the analysis of any type of system described by linear time-invariant difference equations. For example, the z -transform is used in discrete probability problems, and for this case the numbers in the sequence $\{e(k)\}$ are discrete probabilities [3].

Equation (2-7) is the definition of the single-sided z -transform. The double-sided z -transform, sometimes called the generating function [4], is defined as

$$G[\{f(k)\}] = \sum_{k=-\infty}^{\infty} f(k)z^{-k} \quad (2-9)$$

Throughout this book, only the single-sided z -transform as defined in (2-7) will be used, and this transform will be referred to as the ordinary z -transform. If the sequence $e(k)$ is generated from a time function $e(t)$ by sampling every T seconds, $e(k)$ is understood to be $e(kT)$ (i.e., the T is dropped for convenience).

Three examples will now be given to illustrate the z -transform.

Example 2.1

Given $E(z)$ below, find $\{e(k)\}$.

$$E(z) = 1 + 3z^{-1} - 2z^{-2} + z^{-4} + \dots$$

We know, then, from (2-7), that the values of the number sequence $\{e(k)\}$ are

$$\begin{array}{ll} e(0) = 1 & e(3) = 0 \\ e(1) = 3 & e(4) = 1 \\ e(2) = -2 & \dots \end{array}$$

Consider now the identity

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \dots, \quad |x| < 1 \quad (2-10)$$

This power series is useful, in some cases, in expressing $E(z)$ in closed form, as will be illustrated in the next two examples.

Example 2.2

Given that $e(k) = 1$ for all k , find $E(z)$. By definition $E(z)$ is

$$E(z) = 1 + z^{-1} + z^{-2} + \dots$$

The closed form of $E(z)$ is obtained from (2-10):

$$E(z) = \frac{1}{1-z^{-1}} = \frac{z}{z-1}, \quad |z^{-1}| < 1 \quad (2-11)$$

Note that $\{e(k)\}$ may be generated by sampling a unit step function. However, there

are many other time functions that have a value of unity every T seconds, and thus all have the same z -transform.

Example 2.3

Given that $e(k) = \epsilon^{-akT}$, find $E(z)$. $E(z)$ can be written in power series form as

$$\begin{aligned} E(z) &= 1 + \epsilon^{-aT} z^{-1} + \epsilon^{-2aT} z^{-2} + \dots \\ &= 1 + (\epsilon^{-aT} z^{-1}) + (\epsilon^{-aT} z^{-1})^2 + \dots \end{aligned}$$

$E(z)$ can be put in closed form by applying (2-10), so that

$$E(z) = \frac{1}{1 - \epsilon^{-aT} z^{-1}} = \frac{z}{z - \epsilon^{-aT}}, \quad |\epsilon^{-aT} z^{-1}| < 1$$

Note that, in this example, $\{e(k)\}$ may be generated by sampling the function $e(t) = \epsilon^{-at}$.

As with the Laplace transform, each z -transform has a region of existence in the complex plane. This region is of importance if an integral is used to find the inverse z -transform [4]. However, if tables are used for both the forward transform and the inverse transform, the region of existence is not of direct importance. Hence we generally will not state the region of existence when a transform is employed.

2.4 PROPERTIES OF THE z-TRANSFORM

Several properties of the z -transform will now be developed. These properties will prove to be useful in the analysis of discrete systems.

Addition and Subtraction

Property. The z -transform of a sum of number sequences is equal to the sum of the z -transforms of the number sequences; that is,

$$\mathcal{Z}[e_1(k) \pm e_2(k)] = E_1(z) \pm E_2(z) \quad (2-12)$$

Proof. From the definition of the z -transform,

$$\begin{aligned} \mathcal{Z}[e_1(k) \pm e_2(k)] &= \sum_{k=0}^{\infty} [e_1(k) \pm e_2(k)] z^{-k} \\ &= \sum_{k=0}^{\infty} e_1(k) z^{-k} \pm \sum_{k=0}^{\infty} e_2(k) z^{-k} = E_1(z) \pm E_2(z) \end{aligned}$$

Multiplication by a Constant

Property. The z -transform of a number sequence multiplied by a constant is equal to the constant multiplied by the z -transform of the number sequence:

$$\mathcal{Z}[ae(k)] = a\mathcal{Z}[e(k)] = aE(z) \quad (2-13)$$

Proof. From the definition of the z-transform,

$$\mathcal{Z}[ae(k)] = \sum_{k=0}^{\infty} ae(k)z^{-k} = a \sum_{k=0}^{\infty} e(k)z^{-k} = aE(z)$$

Properties (2-12) and (2-13) form the *linearity property* of the z-transform.

Example 2.4

The linearity property of the z-transform can also be proved as follows. Let

$$e(k) = ae_1(k) + be_2(k)$$

Then

$$\begin{aligned} \mathcal{Z}[e(k)] &= \mathcal{Z}[ae_1(k) + be_2(k)] \\ &= \sum_{k=0}^{\infty} [ae_1(k) + be_2(k)]z^{-k} \\ &= a \sum_{k=0}^{\infty} e_1(k)z^{-k} + b \sum_{k=0}^{\infty} e_2(k)z^{-k} \\ &= a\mathcal{Z}[e_1(k)] + b\mathcal{Z}[e_2(k)] \\ &= aE_1(z) + bE_2(z) \end{aligned}$$

Real Translation

Property. Let n be a positive integer, and let $E(z)$ be the z-transform of $\{e(k)\}$. Then

$$\mathcal{Z}[e(k-n)u(k-n)] = z^{-n}E(z) \quad (2-14)$$

and

$$\mathcal{Z}[e(k+n)u(k)] = z^n \left[\underline{E(z)} - \sum_{k=0}^{n-1} e(k)z^{-k} \right] \quad (2-15)$$

The inversion integral in the z-transform definition (2-7) yields $e(k) = 0$ for all $k < 0$; hence, the z-transform of $e(k)$ can be expressed as

$$\mathcal{Z}[e(k)] = \mathcal{Z}[e(k)u(k)]$$

where $u(k)$ is the *discrete unit step function* defined by

$$u(k) = \begin{cases} 0, & k < 0 \\ 1, & k \geq 0 \end{cases}$$

The time-delayed function in (2-14) is given by

$$e(k-n)u(k-n) = e(k)u(k)|_{k \leftarrow k-n}$$

In (2-15), advancing $e(k)$ in time results in

$$\begin{aligned}\mathcal{Z}[e(k)u(k)]|_{k \leftarrow k+n} &= \mathcal{Z}[e(k+n)u(k+n)] \\ &= \mathcal{Z}[e(k+n)u(k)]\end{aligned}$$

We now prove (2-14) and (2-15).

Proof. From the definition of the z-transform,

$$\begin{aligned}\mathcal{Z}[e(k-n)u(k-n)] &= e(0)z^{-n} + e(1)z^{-(n+1)} + e(2)z^{-(n+2)} + \dots \\ &= z^{-n}[e(0) + e(1)z^{-1} + e(2)z^{-2} + \dots] = z^{-n}E(z)\end{aligned}$$

which proves (2-14). Also,

$$\mathcal{Z}[e(k+n)u(k)] = e(n) + e(n+1)z^{-1} + e(n+2)z^{-2} + \dots$$

By adding and subtracting terms and factoring z^n , we obtain

$$\begin{aligned}\mathcal{Z}[e(k+n)u(k)] &= z^n[e(0) + e(1)z^{-1} + e(2)z^{-2} + \dots \\ &\quad + e(n-1)z^{-(n-1)} + e(n)z^{-n} + e(n+1)z^{-(n+1)} + e(n+2)z^{-(n+2)} \\ &\quad + \dots - e(0) - e(1)z^{-1} - e(2)z^{-2} - \dots - e(n-1)z^{-(n-1)}]\end{aligned}$$

Or,

$$\mathcal{Z}[e(k+n)u(k)] = z^n \left[E(z) - \sum_{k=0}^{n-1} e(k)z^{-k} \right]$$

which proves (2-15).

To illustrate these properties further, consider the number sequence shown in Table 2-1, which illustrates the effects of shifting by two sample periods. For the sequence $e(k-2)u(k-2)$, no numbers of the sequence $e(k)$ are lost; thus the z-transform of $e(k-2)u(k-2)$ can be expressed as a simple function of $E(z)$. However, in forming the sequence $e(k+2)u(k)$, the first two values of $e(k)$ have been lost, and the z-transform of $e(k+2)u(k)$ cannot be expressed as a simple function of $E(z)$.

TABLE 2-1 EXAMPLES OF SHIFTING

k	$e(k)$	$e(k-2)$	$e(k+2)$
0	2	0	1.3
1	1.6	0	1.1
2	1.3	2	1.0
3	1.1	1.6	...
4	1.0	1.3	...
...

Example 2.5

It was shown in Example 2.3 that

$$\mathcal{Z}[\epsilon^{-akT}] = \frac{z}{z - \epsilon^{-aT}}$$

Thus

$$\mathcal{Z}[\epsilon^{-a(k-3)T} u[(k-3)T]] = z^{-3} \left[\frac{z}{z - \epsilon^{-aT}} \right] = \frac{1}{z^2(z - \epsilon^{-aT})}$$

where $u(kT)$ is the unit step. Also,

$$\mathcal{Z}[\epsilon^{-a(k+2)T} u(kT)] = z^2 \left[\frac{z}{z - \epsilon^{-aT}} - 1 - \epsilon^{-aT} z^{-1} \right]$$

We now define the *discrete unit impulse function* $\delta(k - n)$:

$$\delta(k - n) = \begin{cases} 1, & k = n \\ 0, & k \neq n \end{cases} \quad (2-16)$$

This function is also called the *unit sample function*. The z-transform of the unit impulse function is then, for $n \geq 0$,

$$\mathcal{Z}[\delta(k - n)] = \sum_{k=0}^{\infty} \delta(k - n) z^{-k} = \delta(k - n) z^{-n} = z^{-n}$$

from (2-16).

Complex Translation

Property. Given that the z-transform of $e(k)$ is $E(z)$. Then

$$\mathcal{Z}[\epsilon^{ak} e(k)] = E(z\epsilon^{-a}) \quad (2-17)$$

Proof. From the definition of the z-transform,

$$\begin{aligned} \mathcal{Z}[\epsilon^{ak} e(k)] &= e(0) + \epsilon^a e(1)z^{-1} + \epsilon^{2a} e(2)z^{-2} + \dots \\ &= e(0) + e(1)(z\epsilon^{-a})^{-1} + e(2)(z\epsilon^{-a})^{-2} + \dots \end{aligned}$$

or

$$\mathcal{Z}[\epsilon^{ak} e(k)] = E(z)|_{z \leftarrow z\epsilon^{-a}} = E(z\epsilon^{-a})$$

Example 2.6

Given that the z-transform of $e(k) = k$ is $E(z) = z/(z - 1)^2$, then the z-transform of $k\epsilon^{ak}$ is

$$\begin{aligned} E(z)|_{z \leftarrow z\epsilon^{-a}} &= \frac{z}{(z - 1)^2} \Big|_{z \leftarrow z\epsilon^{-a}} \\ &= \frac{z\epsilon^{-a}}{(z\epsilon^{-a} - 1)^2} = \frac{\epsilon^a z}{(z - \epsilon^a)^2} \end{aligned}$$

Initial Value

Property. Given that the z-transform of $e(k)$ is $E(z)$. Then

$$e(0) = \lim_{z \rightarrow \infty} E(z) \quad (2-18)$$

Proof. Since

$$E(z) = e(0) + e(1)z^{-1} + e(2)z^{-2} + \dots$$

then (2-18) is seen by inspection.

Final Value

Property. Given that the z-transform of $e(k)$ is $E(z)$. Then

$$\lim_{n \rightarrow \infty} e(n) = \lim_{z \rightarrow 1} (z - 1)E(z)$$

provided that the left-side limit exists.

Proof. Consider the transform

$$\begin{aligned} \mathcal{Z}[e(k+1) - e(k)] &= \lim_{n \rightarrow \infty} \left[\sum_{k=0}^n e(k+1)z^{-k} - \sum_{k=0}^n e(k)z^{-k} \right] \\ &= \lim_{n \rightarrow \infty} [-e(0) + e(1)(1 - z^{-1}) + e(2)(z^{-1} - z^{-2}) + \dots \\ &\quad + e(n)(z^{-n+1} - z^{-n}) + e(n+1)z^{-n}] \end{aligned}$$

Thus

$$\lim_{z \rightarrow 1} [\mathcal{Z}[e(k+1) - e(k)]] = \lim_{n \rightarrow \infty} [e(n+1) - e(0)]$$

Also, from the real translation property,

$$\begin{aligned} \mathcal{Z}[e(k+1) - e(k)] &= z[E(z) - e(0)] - E(z) \\ &= (z - 1)E(z) - ze(0) \end{aligned}$$

Equating the two expressions above, we obtain

$$\lim_{n \rightarrow \infty} e(n) = \lim_{z \rightarrow 1} (z - 1)E(z)$$

provided that the left-side limit exists. It is shown in Chapter 7 that this limit exists provided that all poles of $E(z)$ are inside the unit circle, except for possibly a simple pole at $z = 1$.

Example 2.7

To illustrate the initial-value property and the final-value property, consider the z-transform of $e(k) = 1, k = 0, 1, 2, \dots$. We have shown, in Example 2.2, that

$$E(z) = \mathcal{Z}[1] = \frac{z}{z - 1}$$

Applying the initial-value property, we see that

$$e(0) = \lim_{z \rightarrow \infty} \frac{z}{z-1} = \lim_{z \rightarrow \infty} \frac{1}{1-1/z} = 1$$

Since the final value of $e(k)$ exists, we may apply the final-value property.

$$\lim_{k \rightarrow \infty} e(k) = \lim_{z \rightarrow 1} (z-1)E(z) = \lim_{z \rightarrow 1} z = 1$$

The derived properties of the z -transform are listed in Table 2-2, which also includes additional properties. These additional properties are derived later, or else the derivations are given as problems. The notation $e_1(k) * e_2(k)$ indicates convolution, and is discussed in Section 2.6.

TABLE 2-2 PROPERTIES OF THE z -TRANSFORM

Sequence	Transform
$e(k)$	$E(z) = \sum_{k=0}^{\infty} e(k)z^{-k}$
$a_1 e_1(k) + a_2 e_2(k)$	$a_1 E_1(z) + a_2 E_2(z)$
$e(k-n)u(k-n); \quad n \geq 0$	$z^{-n} E(z)$
$e(k+n)u(k); \quad n \geq 1$	$z^n \left[E(z) - \sum_{k=0}^{n-1} e(k)z^{-k} \right]$
$\epsilon^{ak} e(k)$	$E(z\epsilon^{-a})$
$ke(k)$	$-z \frac{dE(z)}{dz}$
$e_1(k) * e_2(k)$	$E_1(z)E_2(z)$
$e_1(k) = \sum_{n=0}^k e(n)$	$E_1(z) = \frac{z}{z-1} E(z)$
Initial value: $e(0) = \lim_{z \rightarrow \infty} E(z)$	
Final value: $e(\infty) = \lim_{z \rightarrow 1} (z-1)E(z)$, if $e(\infty)$ exists	

2.5 SOLUTION OF DIFFERENCE EQUATIONS

There are three basic techniques for solving linear time-invariant difference equations. The first method, commonly referred to as the classical approach, consists of finding the complementary and the particular parts of the solution [6], in a manner similar to that used in the classical solution of linear differential equations. This technique will not be discussed here; however, a technique similar to it is presented later in this chapter in the discussion of state variables. The second technique, which is a sequential procedure, is the method used in the digital-computer solution of

difference equations and is illustrated by the following example. The third technique will be considered later.

Example 2.8



It is desired to find $m(k)$ for the equation

$$m(k) = e(k) - e(k-1) - m(k-1), \quad k \geq 0$$

where

$$e(k) = \begin{cases} 1, & k \text{ even} \\ 0, & k \text{ odd} \end{cases}$$

and both $e(-1)$ and $m(-1)$ are zero. Then $m(k)$ can be determined by solving the difference equation first for $k = 0$, then for $k = 1, k = 2$, and so on. Thus

$$m(0) = e(0) - e(-1) - m(-1) = 1 - 0 - 0 = 1$$

$$m(1) = e(1) - e(0) - m(0) = 0 - 1 - 1 = -2$$

$$m(2) = e(2) - e(1) - m(1) = 1 - 0 + 2 = 3$$

$$m(3) = e(3) - e(2) - m(2) = 0 - 1 - 3 = -4$$

$$m(4) = e(4) - e(3) - m(3) = 1 - 0 + 4 = 5$$

Note the sequential nature of the solution process in Example 2.8. Using this approach, we can find $m(k)$ for any value of k . This technique is not practical, however, for large values of k , except when implemented on a digital computer. For the example above, a program segment, in MATLAB, which solves the equation for $0 \leq k \leq 20$ is

```
mkminus1 = 0;
ekminus1 = 0;
ek = 1;
for k=0:20
    mk = ek - ekminus1 - mkminus1;
    [k,mk]
    mkminus1 = mk;
    ekminus1 = ek;
    ek = 1 - ek;
end
```

In the coding, $mkminus1$ represents $m(k-1)$, mk represents $m(k)$, $ekminus1$ represents $e(k-1)$, and ek represents $e(k)$. The first three MATLAB statements initialize the sequential process, which begins with the fourth statement and extends to the last statement. A loop exists between the fourth and ninth statements and the first time through the loop $mk = m(0)$, the second time through $mk = m(1)$, and so on.

The third technique for solving linear time-invariant difference equations, which employs the use of the z -transform, will now be presented. Consider the following n th-order difference equation, where it is assumed that $\{e(k)\}$ is known.

$$\begin{aligned} m(k) + a_{n-1}m(k-1) + \cdots + a_0m(k-n) \\ = b_n e(k) + b_{n-1}e(k-1) + \cdots + b_0e(k-n) \end{aligned} \quad (2-19)$$

The z -transform of (2-19), which results from the use of the real translation property (2-14), is

$$\begin{aligned} M(z) + a_{n-1}z^{-1}M(z) + \cdots + a_0z^{-n}M(z) \\ = b_n E(z) + b_{n-1}z^{-1}E(z) + \cdots + b_0z^{-n}E(z) \end{aligned} \quad (2-20)$$

Note that the z -transform has changed the difference equation in (2-19) to the algebraic equation in (2-20). Solving the expression above for $M(z)$ yields

$$M(z) = \frac{b_n + b_{n-1}z^{-1} + \cdots + b_0z^{-n}}{1 + a_{n-1}z^{-1} + \cdots + a_0z^{-n}} E(z) \quad (2-21)$$

$m(k)$ can be found by taking the inverse z -transform of (2-21). General techniques for determining the inverse z -transform are discussed in the next section.

Example 2.9

Consider the difference equation of Example 2.8.

$$m(k) = e(k) - e(k-1) - m(k-1)$$

The z -transform of this equation, obtained via the real translation property, is

$$M(z) = E(z) - z^{-1}E(z) - z^{-1}M(z)$$

or

$$M(z) = \frac{z-1}{z+1} E(z)$$

Since

$$e(k) = \begin{cases} 1, & k \text{ even} \\ 0, & k \text{ odd} \end{cases}$$

we see that

$$E(z) = 1 + z^{-2} + z^{-4} + \cdots = \frac{1}{1 - z^{-2}} = \frac{z^2}{z^2 - 1} = \frac{z^2}{(z-1)(z+1)}$$

Thus

$$M(z) = \frac{z-1}{z+1} \frac{z^2}{(z-1)(z+1)} = \frac{z^2}{z^2 + 2z + 1}$$

We can expand $M(z)$ into a power series by dividing the numerator of $M(z)$ by its

denominator so as to obtain

$$\begin{array}{r}
 1 - 2z^{-1} + 3z^{-2} - 4z^{-3} + \dots \\
 z^2 + 2z + 1 \overline{) z^2} \\
 \underline{z^2 + 2z + 1} \\
 -2z - 1 \\
 \underline{-2z - 4 - 2z^{-1}} \\
 3 + 2z^{-1} \\
 \underline{3 + 6z^{-1} + 3z^{-2}} \\
 -4z^{-1} - 3z^{-2} \\
 \dots
 \end{array}$$

(We discuss this method for finding the inverse transform in the next section.)
Therefore,

$$M(z) = 1 - 2z^{-1} + 3z^{-2} - 4z^{-3} + \dots$$

and the values of $m(k)$ are seen to be the same as those found using the sequential technique in Example 2.8.

Thus far we have considered only difference equations for which the initial conditions are zero. The solution presented in (2-21) represents only the forced part of the response. In order to include initial conditions in the solution of (2-19), k is replaced with $k + n$. Then

$$\begin{aligned}
 m(k + n) + a_{n-1}m(k + n - 1) + \dots + a_0m(k) \\
 = b_n e(k + n) + b_{n-1}e(k + n - 1) + \dots + b_0 e(k)
 \end{aligned} \quad (2-22)$$

From the real translation property (2-15),

$$\mathcal{Z}[m(k + i)] = z^i [M(z) - m(0) - m(1)z^{-1} - \dots - m(i - 1)z^{-(i-1)}] \quad (2-23)$$

Thus the z-transform of (2-22) can be found using (2-23), and all initial conditions are included in the solution. Note that if all initial conditions are zero, the z-transform of (2-22) yields the same results as given in (2-21). Note also that for the n th order difference equation (2-22), the initial conditions are, from (2-23), $m(0), m(1), \dots, m(n - 1)$. Thus the term *initial conditions* has a different meaning than for a differential equation.

2.6 THE INVERSE z-TRANSFORM

In order for the z-transform technique to be a feasible approach in the solution of difference equations, methods for determining the inverse z-transform are required. Four such methods will be given here.

Power Series Method

The power series method for finding the inverse z-transform of a function $E(z)$ which is expressed as the ratio of two polynomials in z involves dividing the denominator

of $E(z)$ into the numerator such that a power series of the form

$$E(z) = e_0 + e_1 z^{-1} + e_2 z^{-2} + \dots \quad (2-24)$$

is obtained. From the definition of the z-transform, it can be seen that the values of $\{e(k)\}$ are simply the coefficients in the power series. This technique was illustrated in Example 2.9. Another example will now be given.

Example 2.10

It is desired to find the values of $e(k)$ for $E(z)$ given by the expression

$$E(z) = \frac{z}{z^2 - 3z + 2}$$

Using long division, we obtain

$$\begin{array}{r} z^{-1} + 3z^{-2} + 7z^{-3} + 15z^{-4} + \dots \\ z^2 - 3z + 2 \overline{)z} \\ \underline{z - 3 + 2z^{-1}} \\ 3 - 2z^{-1} \\ \underline{3 - 9z^{-1} + 6z^{-2}} \\ 7z^{-1} - 6z^{-2} \\ \underline{7z^{-1} - 21z^{-2} + 14z^{-3}} \\ 15z^{-2} - 14z^{-3} + \dots \\ \dots\dots\dots \end{array}$$

and therefore

$$\begin{array}{ll} e(0) = 0 & e(4) = 15 \\ e(1) = 1 & \dots \\ e(2) = 3 & e(k) = 2^k - 1 \\ e(3) = 7 & \dots \end{array}$$

In this particular case, the general expression for $e(k)$ as a function of k [i.e., $e(k) = 2^k - 1$] can be recognized. In general, this cannot be done using the power series method.

Partial-Fraction Expansion Method

In a manner similar to that employed with the Laplace transform, a function $E(z)$ can be expanded in partial fractions and then tables of known z-transform pairs can be used to determine the inverse z-transform. A table of z-transforms is given in Table 2-3, and a table of z-transforms based on sampled time functions is given in Appendix VIII. Before proceeding with an example of the partial-fraction expansion method, consider the function

$$E(z) = \frac{z}{z - a} = 1 + az^{-1} + a^2 z^{-2} + a^3 z^{-3} + \dots \quad (2-25)$$

TABLE 2-3 z-TRANSFORMS

Sequence	z-Transform
$\delta(k - n)$	z^{-n}
1	$\frac{z}{z - 1}$
k	$\frac{z}{(z - 1)^2}$
k^2	$\frac{z(z + 1)}{(z - 1)^3}$
a^k	$\frac{z}{z - a}$
ka^k	$\frac{az}{(z - a)^2}$
$\sin ak$	$\frac{z \sin a}{z^2 - 2z \cos a + 1}$
$\cos ak$	$\frac{z(z - \cos a)}{z^2 - 2z \cos a + 1}$
$a^k \sin bk$	$\frac{az \sin b}{z^2 - 2az \cos b + a^2}$
$a^k \cos bk$	$\frac{z^2 - az \cos b}{z^2 - 2az \cos b + a^2}$

Examination of the power series indicates that

$$\mathcal{Z}^{-1}\left[\frac{z}{z - a}\right] = a^k \quad (2-26)$$

where $\mathcal{Z}^{-1}[\cdot]$ indicates the inverse z-transform. This particular function is perhaps the most common z-transform encountered, since the sequence $\{a^k\}$ is exponential in nature (see Problem 2-2).

It is seen from the transform table in Appendix VIII that a factor of z appears in the numerator of the transforms given. Hence the partial-fraction expansion should be performed on $E(z)/z$, which will result in the terms of $E(z)$ being of the same form as those in the tables.

Example 2.11

Consider the function $E(z)$ given in Example 2.10:

$$E(z) = \frac{z}{(z - 1)(z - 2)}$$

Hence

$$\frac{E(z)}{z} = \frac{1}{(z-1)(z-2)} = \frac{-1}{z-1} + \frac{1}{z-2}$$

Then

$$\mathcal{Z}^{-1}[E(z)] = \mathcal{Z}^{-1}\left[\frac{-z}{z-1}\right] + \mathcal{Z}^{-1}\left[\frac{z}{z-2}\right]$$

From (2-26) or Table 2-3, the value of $e(k)$ is given by

$$e(k) = -1 + 2^k$$

which is the same value as that found in Example 2.10.

Consider next the function

$$E_1(z) = z^{-1}E(z) = \frac{1}{(z-1)(z-2)}$$

From the real translation property (2-14), $e_1(k)$ is given by

$$\begin{aligned} e_1(k) &= \mathcal{Z}^{-1}[z^{-1}E(z)] = e(k-1)u(k-1) \\ &= [-1 + 2^{(k-1)}]u(k-1) \\ &= \begin{cases} 0, & k = 0 \\ -1 + 2^{(k-1)}, & k \geq 1 \end{cases} \end{aligned}$$

This inverse can also be found by partial-fraction expansion.

$$\frac{E_1(z)}{z} = \frac{1}{z(z-1)(z-2)} = \frac{\frac{1}{2}}{z} + \frac{-1}{z-1} + \frac{\frac{1}{2}}{z-2}$$

Or,

$$E_1(z) = \frac{1}{2} + \frac{-z}{z-1} + \frac{(\frac{1}{2})z}{z-2}$$

Thus

$$e_1(k) = a - 1 + (\frac{1}{2})(2)^k = a - 1 + (2)^{k-1}$$

where

$$a = \begin{cases} \frac{1}{2}, & k = 0 \\ 0, & k \geq 1 \end{cases}$$

since from Table 2-3,

$$\mathcal{Z}^{-1}\left[\frac{1}{2}\right] = \frac{1}{2}\delta(k) = \begin{cases} \frac{1}{2}, & k = 0 \\ 0, & k \geq 1 \end{cases}$$

Hence the two procedures yield the same results for $e_1(k)$.

Thus far we have considered the inverse transform by partial fractions only for functions that have real poles. The same partial-fraction procedure applies for

complex poles; however, the resulting inverse transform contains complex functions. Of course, the sum of these functions is real. We now develop a different procedure that expresses the inverse transforms as real functions.

First, consider the real function

$$\begin{aligned} y(k) &= A\epsilon^{akT} \cos(bkT + \theta) = \frac{A\epsilon^{akT}}{2} [\epsilon^{jbkT} \epsilon^{j\theta} + \epsilon^{-jbkT} \epsilon^{-j\theta}] \\ &= \frac{A}{2} [\epsilon^{(aT + jbT)k} \epsilon^{j\theta} + \epsilon^{(aT - jbT)k} \epsilon^{-j\theta}] \end{aligned} \quad (2-27)$$

where a and b are real. Euler's relation, given by

$$\cos x = \frac{\epsilon^{jx} + \epsilon^{-jx}}{2}$$

is used in (2-27). The z-transform of this function is given by, from Appendix VIII,

$$\begin{aligned} Y(z) &= \frac{A}{2} \left[\frac{\epsilon^{j\theta} z}{z - \epsilon^{aT + jbT}} + \frac{\epsilon^{-j\theta} z}{z - \epsilon^{aT - jbT}} \right] \\ &= \frac{(A\epsilon^{j\theta}/2)z}{z - \epsilon^{aT + jbT}} + \frac{(A\epsilon^{-j\theta}/2)z}{z - \epsilon^{aT - jbT}} = \frac{k_1 z}{z - p_1} + \frac{k_1^* z}{z - p_1^*} \end{aligned} \quad (2-28)$$

where the asterisk indicates the complex conjugate.

The usual partial-fraction expansion yields terms in the form of (2-28). Hence, given the partial-fraction coefficient k_1 and the pole p_1 in (2-28), we can solve for the discrete-time function of (2-27) using the following relationship from (2-28):

$$p_1 = \epsilon^{aT} \epsilon^{jbT} = \epsilon^{aT} \angle bT \Rightarrow aT = \ln|p_1|; \quad bT = \arg p_1 \quad (2-29)$$

and

$$k_1 = \frac{A\epsilon^{j\theta}}{2} = \frac{A}{2} \angle \theta \Rightarrow A = 2|k_1|; \quad \theta = \arg k_1 \quad (2-30)$$

Hence we calculate aT and bT from the poles, and A and θ from the partial-fraction expansion. We can then express the inverse transform as the sinusoid of (2-27). An illustrative example is given next.

Example 2.12



We find the inverse z-transform of the function

$$\begin{aligned} Y(z) &= \frac{-3.894z}{z^2 + 0.6065} = \frac{-3.894z}{(z - j0.7788)(z + j0.7788)} \\ &= \frac{k_1 z}{z - j0.7788} + \frac{k_1^* z}{z + j0.7788} \end{aligned}$$

Dividing both sides by z , we calculate k_1 :

$$k_1 = (z - j0.7788) \left[\frac{-3.894}{(z - j0.7788)(z + j0.7788)} \right]_{z = j0.7788}$$

$$= \frac{-3.894}{z + j0.7788} \Big|_{z=j0.7788} = \frac{-3.894}{2(j0.7788)} = 2.5 \angle 90^\circ$$

From (2-29) and (2-30), with $p_1 = j0.7788$,

$$aT = \ln |p_1| = \ln(0.7788) = -0.250; \quad bT = \arg p_1 = \pi/2$$

$$A = 2|k_1| = 2(2.5) = 5; \quad \theta = \arg k_1 = \pi/2$$

Hence, from (2-27),

$$\begin{aligned} y(k) &= A\epsilon^{akT} \cos(bkT + \theta) \\ &= 5\epsilon^{-0.25k} \cos\left(\frac{\pi}{2}k + \frac{\pi}{2}\right) = -5\epsilon^{-0.25k} \sin \frac{\pi}{2}k \end{aligned}$$

This result can be verified by finding the z -transform of this function using the table in Appendix VIII.

Note the difficulty in calculating the coefficients in the partial-fraction expansion in Example 2.12. A MATLAB program that will calculate these coefficients is given by

```
num = [0 0 -3.894];
den = [1 0 .6065];
[r,p,k] = residue(num,den)
```

```
results: r =  j2.5      p =  j0.7788
          -j2.5      p = -j0.7788
```

The values of r are the residues (the partial-fraction expansion coefficients) for the corresponding poles p of $Y(z)$, and k has values if the numerator of $Y(z)$ is of higher order than the denominator.

Inversion-Formula Method

Perhaps the most general technique for obtaining the inverse of a z -transform is the inversion integral. This integral, derived via complex variable theory, is, from (2-7),

$$e(k) = \frac{1}{2\pi j} \oint_{\Gamma} E(z) z^{k-1} dz, \quad j = \sqrt{-1} \quad (2-31)$$

This expression is the line integral in the z -plane along the closed path Γ , where Γ is any path that encloses all the finite poles of $E(z)z^{k-1}$ [5,7].

Using the theorem of residues [7], we can evaluate the integral in (2-31) via the expression

$$e(k) = \sum_{\substack{\text{at poles} \\ \text{of } [E(z)z^{k-1}]}} [\text{residues of } E(z)z^{k-1}] \quad (2-32)$$

If the function $E(z)z^{k-1}$ has a simple pole at $z = a$, the residue is evaluated as

$$(\text{residue})_{z=a} = (z - a)E(z)z^{k-1} \Big|_{z=a} \quad (2-33)$$

For a pole of order m at $z = a$, the residue is calculated using the expression

$$(\text{residue})_{z=a} = \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} [(z-a)^m E(z)z^{k-1}] \Big|_{z=a} \quad (2-34)$$

Example 2.13

Consider the function $E(z)$ from Examples 2.10 and 2.11:

$$E(z) = \frac{z}{(z-1)(z-2)}$$

Substituting this expression into (2-32) and (2-33) yields

$$e(k) = \frac{z^k}{z-2} \Big|_{z=1} + \frac{z^k}{z-1} \Big|_{z=2} = -1 + 2^k$$

and the result is seen to be the same as that obtained in the previous examples. As in Example 2.11, let

$$E_1(z) = z^{-1} E(z) = \frac{1}{(z-1)(z-2)}$$

Then, from the real translation property,

$$e_1(k) = e(k-1)u(k-1) = [-1 + 2^{(k-1)}]u(k-1)$$

We can also find $e_1(k)$ by the inversion formula. In (2-32),

$$E_1(z)z^{k-1} = \frac{z^{k-1}}{(z-1)(z-2)}$$

This function has a pole at $z = 0$ only for $k = 0$, and thus

$$\begin{aligned} e_1(0) &= \sum_{\text{at } z=0,1,2} \left[\text{residues of } \frac{1}{z(z-1)(z-2)} \right] \\ &= \frac{1}{2} - 1 + \frac{1}{2} = 0 \end{aligned}$$

The values of $e_1(k)$ for $k \geq 1$ is obtained directly from (2-32), and is left as an exercise for the reader.

The following example illustrates the inversion-formula technique for a multiple-order pole.

Example 2.14

The function $E(z)$ below has a single pole of order 2 at $z = 1$.

$$E(z) = \frac{z}{(z-1)^2}$$

The inverse transform obtained using (2-34) is

$$\begin{aligned}
 e(k) &= \frac{1}{(2-1)!} \frac{d^{2-1}}{dz^{2-1}} \left[(z-1)^2 \left(\frac{z}{(z-1)^2} \right) z^{k-1} \right] \bigg|_{z=1} \\
 &= \frac{d}{dz} (z^k) \bigg|_{z=1} \\
 &= k z^{k-1} \bigg|_{z=1} \\
 &= k
 \end{aligned}$$

Discrete Convolution

The discrete convolution technique used for determining the inverse z-transform is analogous to the convolution integral employed in the use of Laplace transforms. Suppose that the function $E(z)$ can be expressed as the product of two functions, each of which, in general, will be simpler than $E(z)$; that is,

$$E(z) = E_1(z)E_2(z) \quad (2-35)$$

Further, let $E_1(z)$ and $E_2(z)$ be expressed as power series. Then

$$\begin{aligned}
 E(z) &= [e_1(0) + e_1(1)z^{-1} + e_1(2)z^{-2} + \cdots][e_2(0) + e_2(1)z^{-1} \\
 &\quad + e_2(2)z^{-2} + \cdots]
 \end{aligned} \quad (2-36)$$

Direct multiplication of the two power series yields

$$\begin{aligned}
 E(z) &= e_1(0)e_2(0) + [e_1(0)e_2(1) + e_1(1)e_2(0)]z^{-1} \\
 &\quad + [e_1(0)e_2(2) + e_1(1)e_2(1) + e_1(2)e_2(0)]z^{-2} + \cdots
 \end{aligned} \quad (2-37)$$

Thus the general relationship for $e(k)$ is seen to be

$$\begin{aligned}
 e(k) &= e_1(0)e_2(k) + e_1(1)e_2(k-1) + \cdots + e_1(k)e_2(0) \\
 &= \sum_{n=0}^k e_1(n)e_2(k-n) = \sum_{n=0}^k e_1(k-n)e_2(n)
 \end{aligned} \quad (2-38)$$

Equation (2-38) is the discrete convolution summation, and may be useful in determining the inverse z-transform of a function $E(z) = E_1(z)E_2(z)$ if $E_1(z)$ and $E_2(z)$ are initially expressed as power series. Convolution is usually denoted as

$$e(k) = \mathcal{Z}^{-1}[E_1(z)E_2(z)] = e_1(k) * e_2(k) \quad (2-39)$$

Example 2.15

Once again consider the function $E(z)$ from Example 2.10:

$$E(z) = \frac{z}{(z-1)(z-2)} = E_1(z)E_2(z)$$



where we define

$$E_1(z) = \frac{z}{z-1} = 1 + z^{-1} + z^{-2} + \dots$$

and

$$E_2(z) = \frac{1}{z-2} = z^{-1} + 2z^{-2} + (2)^2 z^{-3} + \dots$$

These expansions are obtained by the power-series method. Then $e(k)$ can be formed directly from (2-38). For example, $e(3)$ is given by the expression

$$\begin{aligned} e(3) &= \sum_{n=0}^3 e_1(n)e_2(3-n) \\ &= e_1(0)e_2(3) + e_1(1)e_2(2) + e_1(2)e_2(1) + e_1(3)e_2(0) \\ &= (1)(2^3) + (1)(2^2) + (1)(2) + (1)(1) = 7 \end{aligned}$$

The other values of $e(k)$ can be obtained in a similar manner and a quick check of them will show that they agree with the values obtained via the other inversion techniques. A MATLAB program that performs a discrete convolution for this example is given by

```
e1 = [1  1  1  1  1  1];
e2 = [0  1  2  4  8 16];
e = conv(e1,e2)
```

```
result: 0  1  3  7 15 31
```

This program correctly gives only the first six values of $e(k)$.

2.7 SIMULATION DIAGRAMS AND FLOW GRAPHS

It has been shown that a linear time-variant discrete-time system may be represented by either a difference equation or a transfer function. A third representation commonly used is a simulation diagram. Simulation diagrams for discrete-time systems are presented in this section.

First the basic elements used to construct simulation diagrams for a system described by a linear difference equation are developed. Let the block shown in Figure 2-2a represent a shift register. Every T seconds, a number is shifted into the register, and at that instant, the number that was stored in the register is shifted out. Therefore, if we let $e(k)$ represent the number shifted into the register at $t = kT$, the number shifted out is $e(k-1)$. We let the symbolic representation of this memory device be as shown in Figure 2-2b. This symbol can represent any device that performs the foregoing operation.

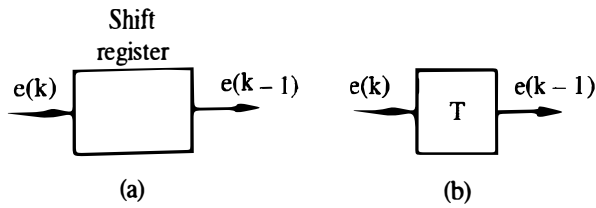


Figure 2-2 Ideal time-delay element.

An interconnection of these devices, together with devices that perform multiplication by a constant and summation, can be used to represent a linear time-invariant difference equation. For example, consider the difference equation used in Example 2.8:

$$m(k) = e(k) - e(k-1) - m(k-1) \quad (2-40)$$

A simulation diagram of this equation is shown in Figure 2-3; writing the equation for the output of the summing junction yields (2-40).

Electronic devices may be constructed to perform all the operations shown in the figure. Now suppose that such a construction exists. Then to solve Example 2.8 using this constructed machine, the numbers in both memory locations (shift registers) are set to zero and the input $e(kT)$ is made equal to 1 at time instants $kT = 0, 2T, 4T, \dots$ and $e(kT)$ is made equal to 0 at time instants $kT = T, 3T, 5T, \dots$. The solution $m(k)$ then appears at the output terminal at $t = kT$. (The reader may calculate the first few values of $m(k)$ to illustrate the validity of the simulation diagram.) This machine would be a special-purpose computer, capable of solving only the difference equation (2-40). Recall that the computer program given below Example 2.8 also solves this difference equation, but in this case a general-purpose computer is used. The general-purpose computer software arranges the arithmetic registers, memory, and so on, to perform the operations depicted in Figure 2-3.

To include nonzero initial conditions in our special-purpose computer described above, replace k with $k + 1$ in (2-40):

$$m(k+1) = e(k+1) - e(k) - m(k)$$

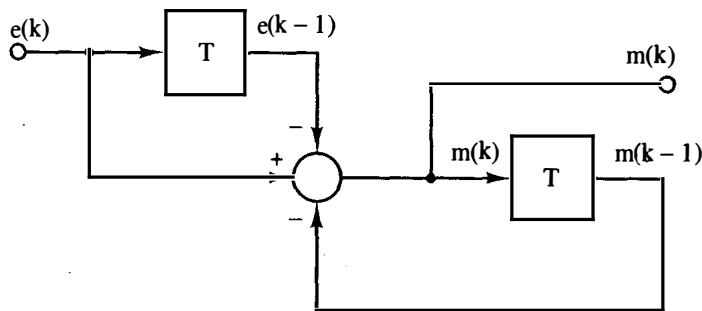


Figure 2-3 Simulation diagram for the difference equation used in Example 2.8.

and in Figure 2-3. Then the value of $e(0)$ is placed in the register, in Figure 2-3, whose output is now $e(k)$, and the value of $m(0)$ is placed in the register whose output is now $m(k)$. The input values $e(1), e(2), \dots$ are applied at the input terminal in Figure 2-3, and the output values $m(1), m(2), \dots$ appear at the output terminal.

Recall that in the analog simulation of continuous systems, the basic element is the integrator. In the simulation of discrete systems, the basic element is the time delay (or memory) of T seconds.

A somewhat different but equivalent graphical representation of a difference equation is the signal flow graph. A block diagram, such as that illustrated in Figure 2-3, is simply a graphical representation of an equation, or a set of equations. The signal flow graph may also be used to graphically represent equations. The basic elements of a flow graph are the branches and the nodes. By definition, the signal out of a branch is equal to the branch gain (transfer function) times the signal into the branch. This is illustrated in Figure 2-4, which shows both the block diagram representation and the flow graph representation of a branch. Also, by definition, the signal at a node in a flow graph is equal to sum of the signals from all branches into that node. This is also shown in Figure 2-4. Thus a flow graph contains exactly the same information as a block diagram.

Once again, consider the time-delay device shown in Figure 2-2b. The z -transform of the input $e(k)$ is $E(z)$, and the z -transform of the output $e(k-1)$ is $z^{-1}E(z)$. Thus the transfer function of the time delay is z^{-1} . (Recall that the transfer function of an integrator is s^{-1} .) Consider again the system shown in Figure 2-3. A flow graph representation of this system is as shown in Figure 2-5. The transfer function of this system may be obtained either from Figure 2-3 by block diagram reduction or from Figure 2-5 by using Mason's gain formula. Those readers unfamiliar with Mason's gain-formula technique are referred to Appendix II. Basically, the gain formula is based on the geometry and signal flow directions of a flow graph. Since this architecture is exactly the same as that used in a block diagram, Mason's gain formula may also be applied directly to a block diagram.

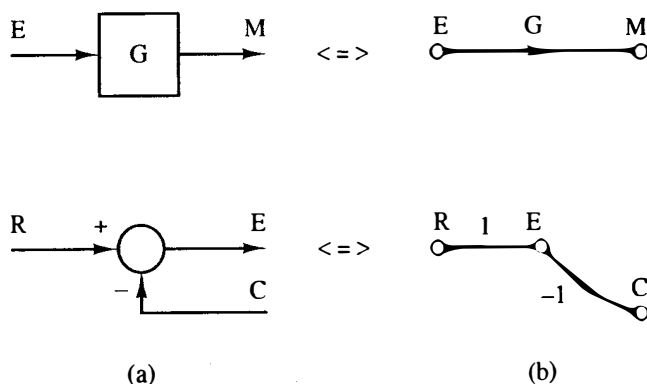


Figure 2-4 Equivalent block diagram (a) and flow graph (b) symbols.

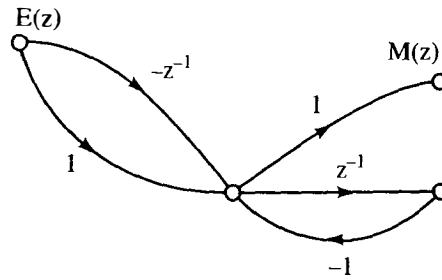


Figure 2-5 Signal flow graph for the system shown in Figure 2-3.

The application of Mason's gain formula to Figure 2-5 yields the transfer function

$$\frac{M(z)}{E(z)} = \frac{1 - z^{-1}}{1 + z^{-1}} = \frac{z - 1}{z + 1}$$

which is the same as that obtained in Example 2.9.

Consider now a general n th-order difference equation:

$$\begin{aligned} m(k) + a_{n-1}m(k-1) + \cdots + a_0m(k-n) \\ = b_n e(k) + b_{n-1}e(k-1) + \cdots + b_0e(k-n) \end{aligned} \quad (2-41)$$

Taking the z -transform of this equation yields

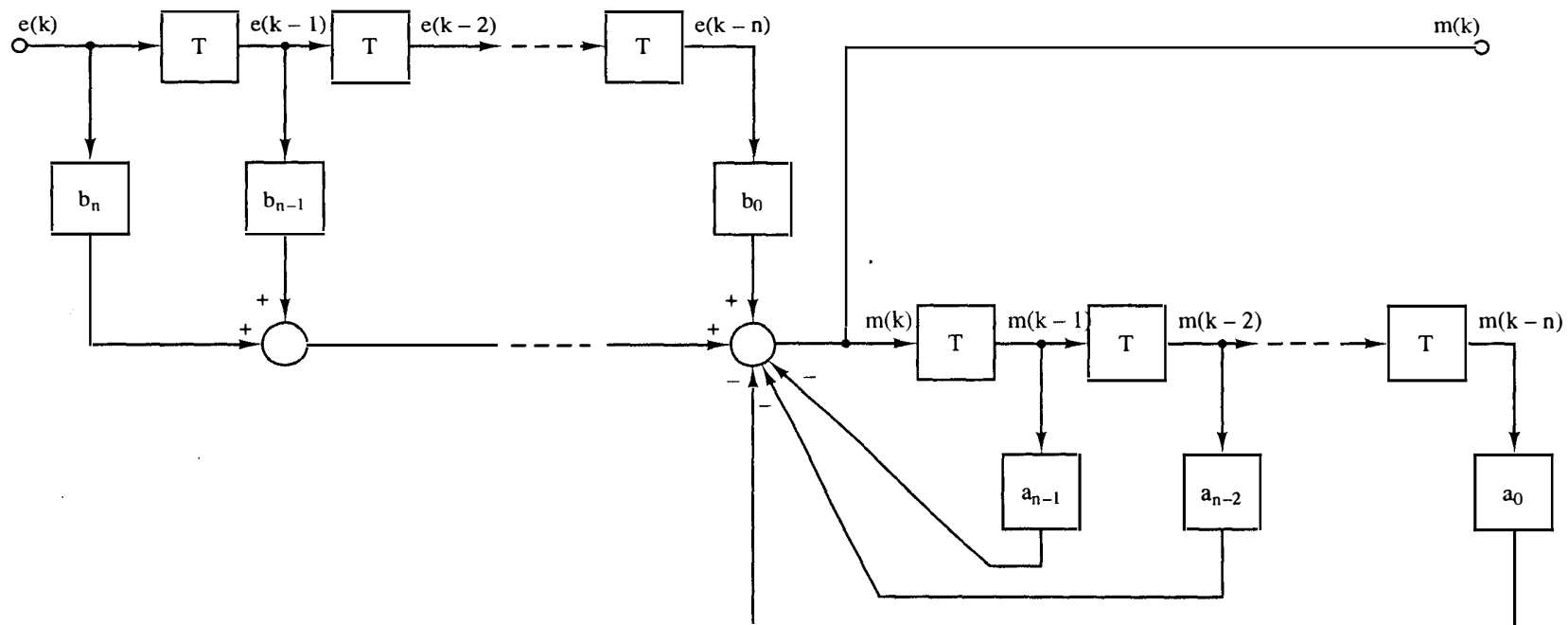
$$\begin{aligned} M(z) + a_{n-1}z^{-1}M(z) + \cdots + a_0z^{-n}M(z) \\ = b_nE(z) + b_{n-1}z^{-1}E(z) + \cdots + b_0z^{-n}E(z) \end{aligned} \quad (2-42)$$

This difference equation may then be represented by the transfer function

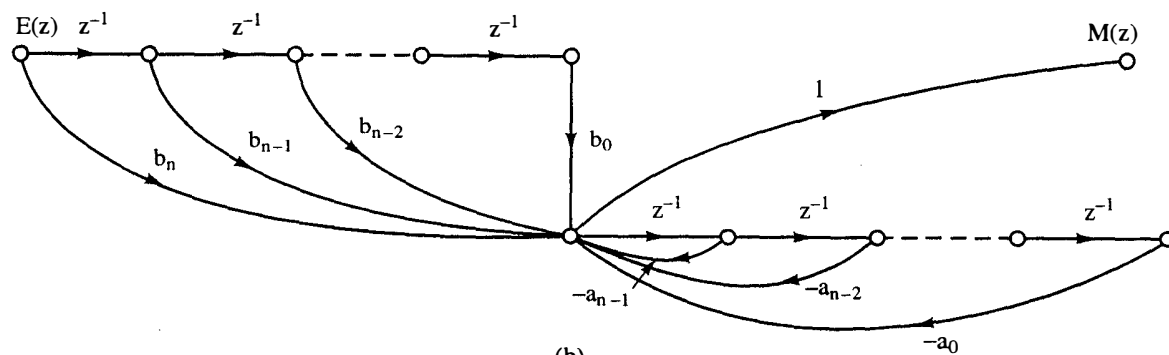
$$\frac{M(z)}{E(z)} = \frac{b_n + b_{n-1}z^{-1} + \cdots + b_0z^{-n}}{1 + a_{n-1}z^{-1} + \cdots + a_0z^{-n}} \quad (2-43)$$

The system of (2-41) may be represented by the simulation diagram shown in Figure 2-6a, since the signal out of the summing junction satisfies (2-41). The flow graph for this diagram is shown in Figure 2-6b. Application of Mason's gain formula to this flow graph yields (2-43).

The simulation diagram of Figure 2-6 is only one of many that can be constructed to represent the transfer function of (2-43). The representation of Figure 2-6 (and the example of Figure 2-3) is nonminimal in the sense that the diagram contains $2n$ delays, but an n th-order system can be represented with only n delays. A minimal representation of the system of Figure 2-3 is given in Figure 2-7. This representation may be verified by calculating the transfer function via Mason's gain formula. This brings us to the topic of state variables for discrete systems, which is



(a)



(b)

Figure 2-6 (a) Simulation diagram for an n th-order difference equation; (b) flow graph for an n th-order difference equation.

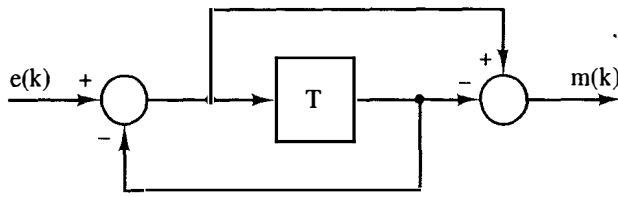


Figure 2-7 A different simulation diagram for Example 2.8.

closely related to simulation diagrams. The remainder of this chapter is devoted to the topic of state variables.

2.8 STATE VARIABLES

In preceding sections we defined a discrete-time system as one that can be described by a difference equation. If the discrete-time system is linear and time invariant, we can also represent the system by a transfer function. For a linear time-invariant discrete-time system with input $E(z)$, output $M(z)$, and transfer function $G(z)$, we can write

$$M(z) = G(z)E(z) \quad (2-44)$$

as illustrated in Section 2.7. Thus the discrete-time system may be represented by the block diagram of Figure 2-8a.

The more modern approach to the analysis and synthesis of discrete-time systems employs what is commonly called the state-variable method. In this approach the system is modeled as shown in Figure 2-8b. To be completely general, we must allow for the possibility of more than one input, and of more than one output. Thus, in Figure 2-8b, the variables $u_i(k)$, $i = 1, \dots, r$, are the external inputs which drive the system; the variables $y_i(k)$, $i = 1, \dots, p$, represent the system outputs or the system responses; and the variables $x_i(k)$, $i = 1, \dots, n$ are the internal or state variables of the system. The state variables represent the minimum amount of information which is necessary to determine both the future states and the system outputs for given input functions; that is, given the system states, the system dynamics and the input functions, we can determine all subsequent states and outputs.

For convenience we represent the system shown in Figure 2-8b by that shown in Figure 2-8c, where $\mathbf{u}(k)$ is the input vector, $\mathbf{y}(k)$ is the output vector, and $\mathbf{x}(k)$ is the state vector:

$$\mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_r(k) \end{bmatrix}, \quad \mathbf{y}(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_p(k) \end{bmatrix}, \quad \mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}$$

The set of values that the input vector $\mathbf{u}(k)$ may assume is called the input space of the system. The output space and state space are defined in a similar manner.

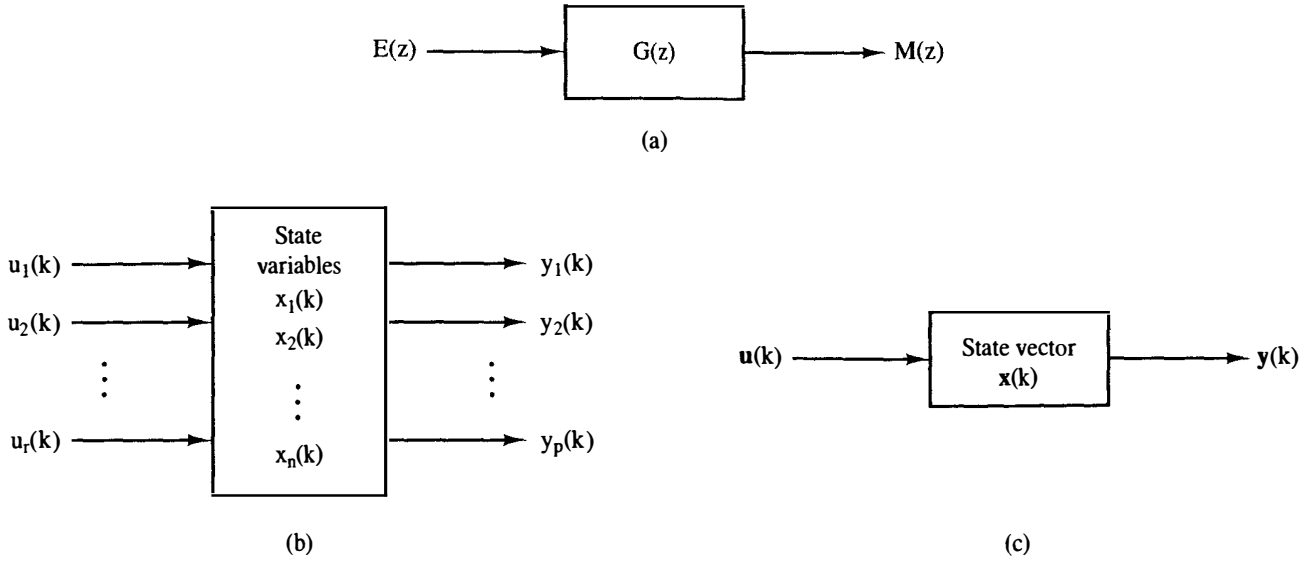


Figure 2-8 Representations of system dynamics: (a) z-transfer function representation; (b) state-variable representation; (c) state-vector representation.

In general, the equation that describes the state of the system at any time $(k + 1)$ is given by the single-valued function relationship

$$\mathbf{x}(k + 1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k)] \quad (2-45)$$

This equation simply states that the state \mathbf{x} at time $(k + 1)$ is a function of the state and the input at the previous discrete-time increment k . The output response of the system is defined in a similar manner as

$$\mathbf{y}(k) = \mathbf{g}[\mathbf{x}(k), \mathbf{u}(k)] \quad (2-46)$$

As stated earlier, when we say time k , we actually mean time kT .

If the system is linear, then equations (2-45) and (2-46) reduce to

$$\mathbf{x}(k + 1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (2-47)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) \quad (2-48)$$

where $\mathbf{x}(k)$ is an n -vector, $\mathbf{u}(k)$ is an r -vector, $\mathbf{y}(k)$ is a p -vector (as shown in Figure 2-8), and $\mathbf{A}(k)$, $\mathbf{B}(k)$, $\mathbf{C}(k)$, and $\mathbf{D}(k)$ are time-varying matrices of dimensions $n \times n$, $n \times r$, $p \times n$, and $p \times r$, respectively. If the system is time invariant, then the matrices in (2-47) and (2-48) are constant, and hence the equations reduce to

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (2-49)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (2-50)$$

(See Appendix IV for a review of matrices.) State-variable modeling will now be illustrated by an example.

Example 2.16

It is desired to find a state-variable model of the system described by the difference equation

$$y(k+2) = u(k) + 1.7y(k+1) - 0.72y(k)$$

Let

$$x_1(k) = y(k)$$

$$x_2(k) = x_1(k+1) = y(k+1)$$

Then

$$x_2(k+1) = y(k+2) = u(k) + 1.7x_2(k) - 0.72x_1(k)$$

or, from these equations, we write

$$x_1(k+1) = x_2(k)$$

$$x_2(k+1) = -0.72x_1(k) + 1.7x_2(k) + u(k)$$

$$y(k) = x_1(k)$$

We may express these equations in vector-matrix form of (2-49) and (2-50) as

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 \\ -0.72 & 1.7 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

Equations (2-49) and (2-50) are the state equations for a linear time-invariant system and usually represent the starting point in the analysis or design of a discrete system by modern methods. However, let us first examine the connection between this approach and the z -transform method. To do this we will give one method for deriving a set of discrete state-variable equations from the z -transform transfer function.

Given the transfer function

$$G(z) = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1z + b_0}{z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0} \quad (2-51)$$

where we have assumed that the order of the numerator is less than that of the denominator. We can write (2-51) as

$$\frac{Y(z)}{U(z)} = G(z) = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1z + b_0}{z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0} \frac{E(z)}{E(z)} \quad (2-52)$$

where the auxiliary variable $E(z)$ has been introduced. We now let

$$Y(z) = (b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1z + b_0)E(z) \quad (2-53)$$

$$U(z) = (z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0)E(z) \quad (2-54)$$

Recall from the real translation property the correspondence

$$\begin{aligned} E(z) &\rightarrow e(k) \\ zE(z) &\rightarrow e(k+1) \\ z^2 E(z) &\rightarrow e(k+2) \\ &\vdots \end{aligned}$$

Under this correspondence we define the state variables

$$\begin{aligned} x_1(k) &= e(k) \\ x_2(k) &= x_1(k+1) = e(k+1) \\ x_3(k) &= x_2(k+1) = e(k+2) \\ &\vdots \\ x_n(k) &= x_{n-1}(k+1) = e(k+n-1) \end{aligned} \quad (2-55)$$

From equations (2-54) and (2-55) we obtain the state equations

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_3(k) \\ x_3(k+1) &= x_4(k) \\ &\vdots \\ x_n(k+1) &= -a_0 x_1(k) - a_1 x_2(k) - a_2 x_3(k) \cdots - a_{n-1} x_n(k) + u(k) \end{aligned} \quad (2-56)$$

which written in matrix form is

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ & & & \vdots & & & \\ -a_0 & -a_1 & -a_2 & -a_3 & -a_4 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(k) \quad (2-57)$$

or simply

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (2-58)$$

The output equation obtained from (2-53) is

$$y(k) = [b_0 \quad b_1 \quad \cdots \quad b_{n-1}] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} \quad (2-59)$$

or simply

$$y(k) = Cx(k) \quad (2-60)$$

Hence equations (2-58) and (2-60) are a set of state equations for the discrete system described by equation (2-51).

Another convenient and useful representation of the discrete system is the signal flow graph or the equivalent simulation diagram. These two forms can be derived from equation (2-52)

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_{n-1}z^{-1} + b_{n-2}z^{-2} + \cdots + b_1z^{1-n} + b_0z^{-n}}{1 + a_{n-1}z^{-1} + \cdots + a_1z^{1-n} + a_0z^{-n}} \frac{E(z)}{E(z)} \quad (2-61)$$

From this expression we obtain the two equations

$$Y(z) = (b_{n-1}z^{-1} + b_{n-2}z^{-2} + \cdots + b_1z^{1-n} + b_0z^{-n})E(z) \quad (2-62)$$

$$U(z) = (1 + a_{n-1}z^{-1} + \cdots + a_1z^{1-n} + a_0z^{-n})E(z) \quad (2-63)$$

Equation (2-63) can be written as

$$E(z) = U(z) - a_{n-1}z^{-1}E(z) - \cdots - a_1z^{1-n}E(z) - a_0z^{-n}E(z) \quad (2-64)$$

A signal flow graph representation of the system described by equation (2-51) can be easily derived from equations (2-62) and (2-64) and is shown in Figure 2-9a.

Recall that the transfer function for a pure delay of T seconds is z^{-1} , where T is the discrete time increment (i.e., T is the interval between sampling instants $k, k+1, k+2$, etc.). Using this relationship, the signal flow graph of Figure 2-9a can be immediately converted to the equivalent simulation diagram of Figure 2-9b. We see from (2-57) and (2-59) that the states are the register outputs, as noted in Figure 2-9b.

Note the relationships that exist between equations (2-57) and (2-59) and the diagrams in Figure 2-9. One who is very familiar with the correspondence between the equations and diagrams can derive a flow graph or simulation diagram from the transfer function by inspection. From the flow graph, one may use Mason's gain formula to reconstruct the original transfer function. The structure of Figure 2-9, together with equations (2-57) and (2-59), is called either the control canonical form or the phase variable canonical form, and is useful in the design procedures presented in Chapters 9 and 10.

Another standard form that is useful in the design procedures of Chapter 9 is the observer canonical form, which is illustrated in Figure 2-10. Note, using Mason's gain formula, that the transfer function for Figure 2-10 is (2-61). It is left as an exercise for the reader to write the state equations for the observer canonical form (see Problem 2-32).

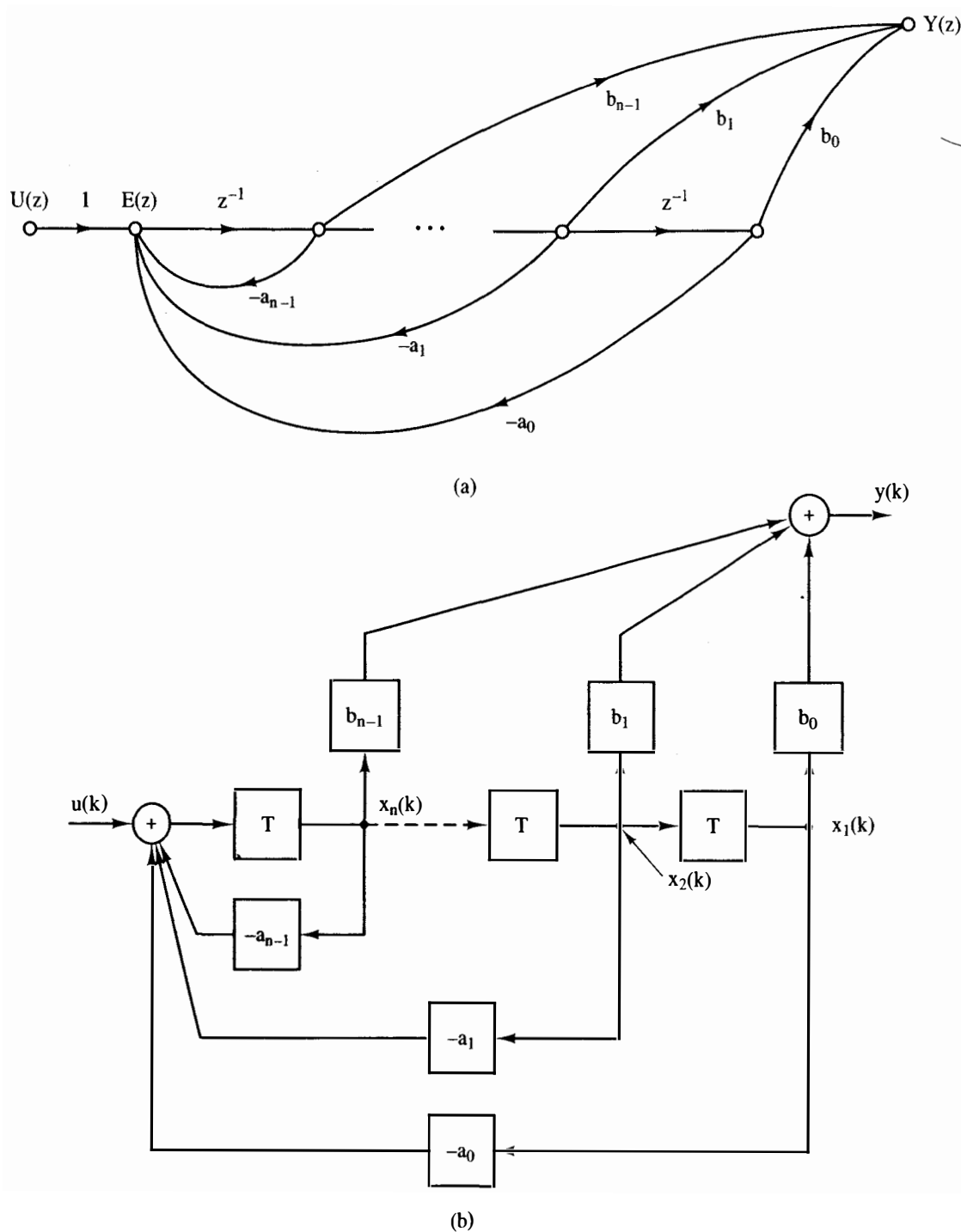


Figure 2-9 Equivalent representations of equation (2-51): (a) signal flow graph representation; (b) simulation diagram.

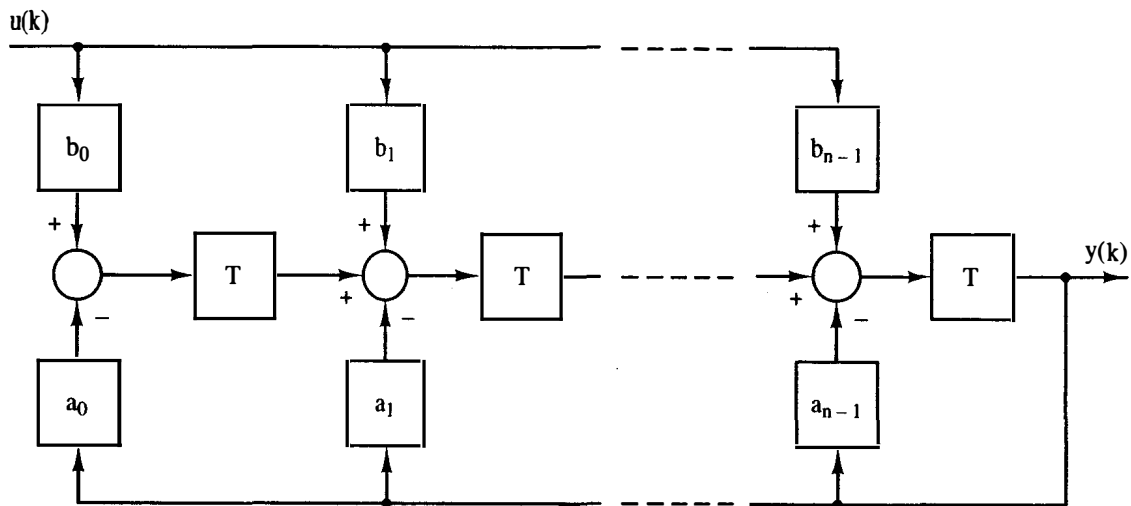


Figure 2-10 Observer canonical form.

Example 2.17

Given the following function, we want to derive a signal flow graph and corresponding state equations:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{z^2 + 2z + 1}{z^3 + 2z^2 + z + \frac{1}{2}}$$

Comparing this expression with (2-51) and Figure 2-9a indicates that the signal flow graph can be derived by inspection, as shown in Figure 2-11. Note that the states are then chosen to be the delay outputs. The state equations can now be derived from Figure 2-11 or from a comparison of the given transfer function with equations (2-51), (2-57), and (2-59). The resulting state equations are

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{1}{2} & -1 & -2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 2 \quad 1] \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}$$

The following example will demonstrate an important salient feature encountered in going from the transfer function to the state equations.

Example 2.18

Given the following transfer function, we wish to derive the state equations.

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_2 z^2 + b_1 z + b_0}{z^2 + a_1 z + a_0}$$

In this case, the order of the numerator is equal to that of the denominator. First we express the transfer function as

$$\frac{Y(z)}{U(z)} = \frac{b_2 + b_1 z^{-1} + b_0 z^{-2}}{1 + a_1 z^{-1} + a_0 z^{-2}} \frac{E(z)}{E(z)}$$

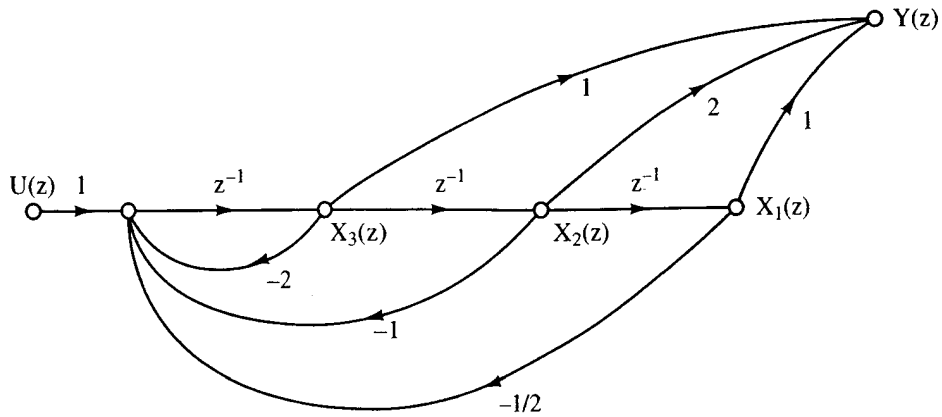


Figure 2-11 Flow graph for Example 2.17.

We see from Mason's gain formula that the signal flow graph for this transfer function is as shown in Figure 2-12. The states are shown on the flow graph, and the equations are seen to be

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

At this point it is important to note that the flow graph in Figure 2-12 is slightly different from the form presented earlier; that is, there is a direct path from $E(z)$ to $Y(z)$. From the flow graph the equation for the output is

$$Y(z) = b_0 X_1(z) + b_1 X_2(z) + b_2 E(z)$$

But

$$E(z) = U(z) - a_0 X_1(z) - a_1 X_2(z)$$

From these equations we obtain

$$Y(z) = b_2 U(z) + (b_0 - b_2 a_0) X_1(z) + (b_1 - b_2 a_1) X_2(z)$$

Hence the output equation is

$$y(k) = [(b_0 - b_2 a_0) \quad (b_1 - b_2 a_1)] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + b_2 u(k)$$

Note the difference in this equation and (2-59).

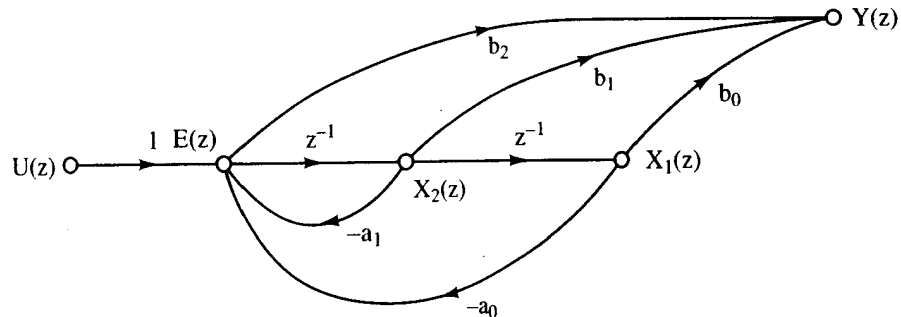


Figure 2-12 Flow graph for Example 2.18.

Whenever the numerator and denominator of the transfer function are of the same order, the output state equation must be handled as demonstrated above. Also, the output equation may be written by inspection from Figure 2-12.

Example 2.19

We wish to derive the state equations for the multivariable discrete system shown in Figure 2-13. The state equation can be obtained directly from the simulation diagram by inspection.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

In summary, state models may be derived from transfer functions by the procedure illustrated in (2-52) through (2-59). A second procedure is:

1. Draw a simulation diagram of the system, using any convenient method.
2. Assign a state variable to each delay output.
3. Write the equation for each delay input and each system output in terms of only the delay outputs and the system input.

The system transfer function gives an input–output description of the system, while the state model gives an internal description in addition to the input–output description. The transfer function for a given system is unique, while the internal model (state model) is not.

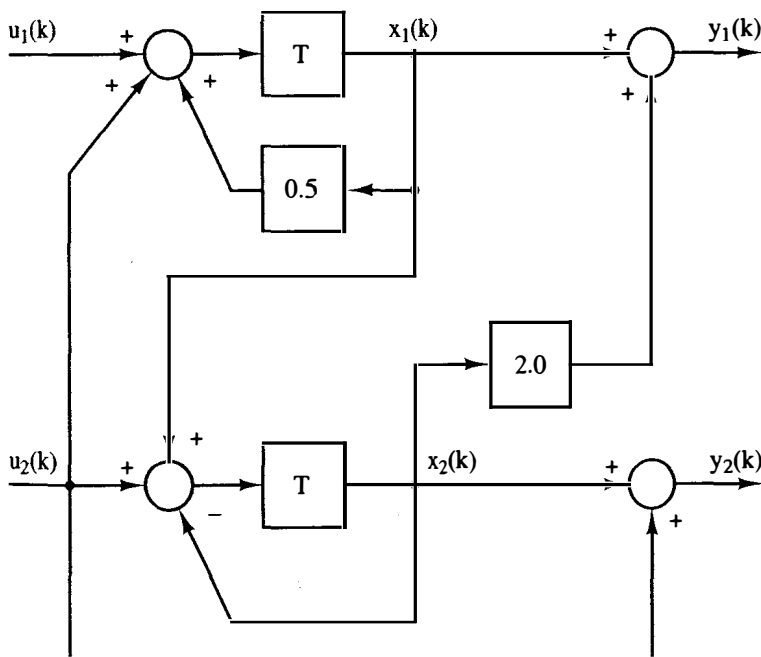


Figure 2-13 Simulation diagram for a multivariable system.

2.9 OTHER STATE-VARIABLE FORMULATIONS

We have seen in the preceding section, techniques for finding a state-variable formulation for a single-input, single-output discrete-time system, given either the system difference equation or the system transfer function. For a given system, there is no unique state-variable formulation. For a given transfer function, any simulation diagram with the system transfer function yields a valid state-variable model for the system. However, for certain analysis or design procedures, certain formulations present advantages with respect to calculations, as will be shown in later chapters. The following example illustrates some derivations of different formulations.

Example 2.20

Consider the difference equation of Example 2.16:

$$y(k + 2) = u(k) + 1.7y(k + 1) - 0.72y(k)$$

One state-variable model was derived in that example. Two additional models are derived here. Taking the z-transform of the difference equation yields

$$\frac{Y(z)}{U(z)} = \frac{1}{z^2 - 1.7z + 0.72}$$

This transfer function can be expressed as

$$\frac{Y(z)}{U(z)} = \left[\frac{1}{z - 0.9} \right] \left[\frac{1}{z - 0.8} \right]$$

A simulation diagram for this transfer function is given in Figure 2-14a. From this figure we write the state equations

$$\begin{aligned} \mathbf{x}(k + 1) &= \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

Also, the system transfer function can be expressed, through partial-fraction expansion, as

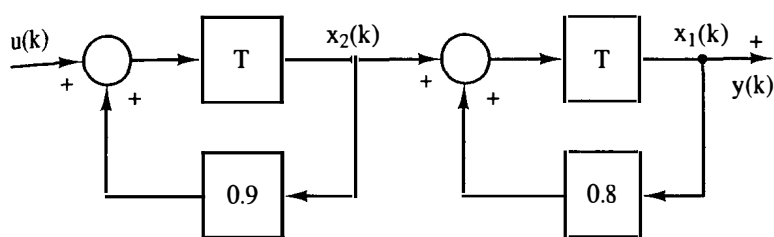
$$\frac{Y(z)}{U(z)} = \frac{10}{z - 0.9} + \frac{-10}{z - 0.8}$$

A simulation diagram for this transfer function is given in Figure 2-14b. From this figure we write the state equations

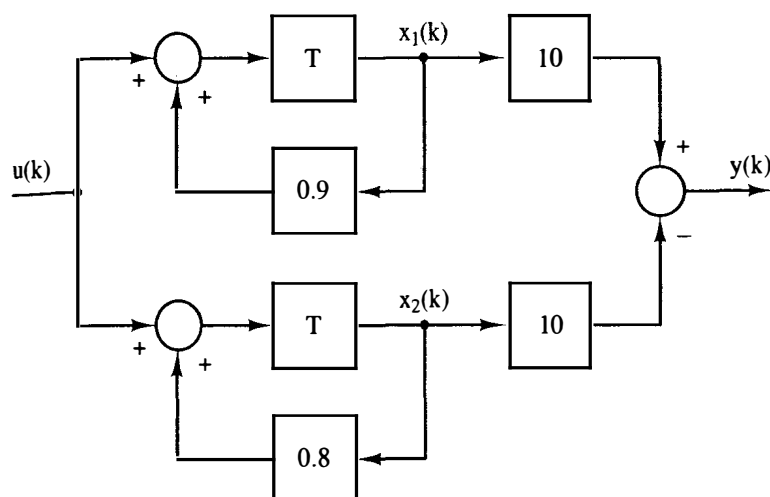
$$\begin{aligned} \mathbf{x}(k + 1) &= \begin{bmatrix} 0.9 & 0 \\ 0 & 0.8 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [10 \quad -10] \mathbf{x}(k) \end{aligned}$$

In Example 2.20, we first represented a higher-order transfer function as a product of simpler transfer functions. For the general case, we would express the transfer function $G(z)$ as

$$G(z) = G_1(z)G_2(z) \cdots G_{n_c}(z) \quad (2-65)$$



(a)



(b)

Figure 2-14 Simulation diagrams for Example 2.20.

Each of the simpler transfer functions $G_{i_c}(z)$ is realized by a technique of the preceding section, and the realizations are then connected in cascade to realize $G(z)$. If $G(z)$ contains either complex poles or complex zeros, we may choose some of the $G_{i_c}(z)$ in (2-65) to be second order, to avoid computational difficulties with complex elements in the state-variable matrices.

Next, in Example 2-20, we represented a higher-order transfer function as the sum of simpler transfer functions through partial-fraction expansion. For the general case, we would express the transfer function $G(z)$ as

$$G(z) = G_{1_p}(z) + G_{2_p}(z) + \cdots + G_{n_p}(z) \quad (2-66)$$

Each of the simpler transfer functions $G_{i_p}(z)$ is realized by a technique of Section 2.8, and the realizations are then connected in parallel to realize $G(z)$.

We have just seen three methods for arriving at different state models for a discrete-time system, with these models derived from the system transfer function. In fact, we can derive any number of different state models, given one state model of the system, through *similarity transformations*. This procedure will now be developed. Consider the state equation in the form

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \quad (2-67)$$

Now, we can apply the linear transformation

$$\mathbf{x}(k) = \mathbf{P}\mathbf{w}(k) \quad (2-68)$$

that is,

$$x_1(k) = p_{11} w_1(k) + p_{12} w_2(k) + \cdots + p_{1n} w_n(k)$$

and so on. Hence \mathbf{P} is a constant ($n \times n$) matrix and $\mathbf{w}(k)$ is the new state vector. Note that it is necessary that \mathbf{P}^{-1} , the inverse of \mathbf{P} , exists so that $\mathbf{w}(k)$ can be determined from $\mathbf{x}(k)$. Substituting (2-68) into (2-67) yields the equations

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{P}^{-1} \mathbf{A} \mathbf{P} \mathbf{w}(k) + \mathbf{P}^{-1} \mathbf{B} \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C} \mathbf{P} \mathbf{w}(k) + \mathbf{D} \mathbf{u}(k) \end{aligned} \quad (2-69)$$

These equations can be expressed as

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{A}_w \mathbf{w}(k) + \mathbf{B}_w \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_w \mathbf{w}(k) + \mathbf{D}_w \mathbf{u}(k) \end{aligned} \quad (2-70)$$

where

$$\begin{aligned} \mathbf{A}_w &= \mathbf{P}^{-1} \mathbf{A} \mathbf{P}, & \mathbf{B}_w &= \mathbf{P}^{-1} \mathbf{B} \\ \mathbf{C}_w &= \mathbf{C} \mathbf{P}, & \mathbf{D}_w &= \mathbf{D} \end{aligned} \quad (2-71)$$

Thus for each different \mathbf{P} for which \mathbf{P}^{-1} exists, a different state model of a given system can be found.

The *characteristic equation* of a matrix \mathbf{A} is defined by the determinant [8]

$$|z\mathbf{I} - \mathbf{A}| = 0 \quad (2-72)$$

and the *characteristic values* (or *eigenvalues*) of the matrix are the roots of the characteristic equation; that is, z_i are the characteristic values of \mathbf{A} if

$$|z\mathbf{I} - \mathbf{A}| = (z - z_1)(z - z_2) \cdots (z - z_n) = 0 \quad (2-73)$$

It will be shown in Chapter 7 that (2-73) determines system stability for the system of (2-67). Note that the characteristic equation of the system matrix is unchanged through the linear transformation,

$$\begin{aligned} |z\mathbf{I} - \mathbf{A}_w| &= |z\mathbf{I} - \mathbf{P}^{-1} \mathbf{A} \mathbf{P}| = |z\mathbf{P}^{-1} \mathbf{I} \mathbf{P} - \mathbf{P}^{-1} \mathbf{A} \mathbf{P}| \\ &= |\mathbf{P}^{-1} \|z\mathbf{I} - \mathbf{A} \| \mathbf{P}| \\ &= |z\mathbf{I} - \mathbf{A}| \end{aligned} \quad (2-74)$$

A linear transformation of the form

$$\mathbf{A}_w = \mathbf{P}^{-1} \mathbf{A} \mathbf{P} \quad (2-75)$$

which is called a *similarity transformation*, has the following properties:

1. As shown above, the characteristic values of the matrix are unchanged under the transformation.
2. The determinant of \mathbf{A}_w (which is the product of characteristic values) is equal to the determinant of \mathbf{A} :

$$|\mathbf{A}_w| = |\mathbf{P}^{-1}\mathbf{A}\mathbf{P}| = |\mathbf{P}^{-1}||\mathbf{A}||\mathbf{P}| = |\mathbf{A}| = z_1 z_2 \cdots z_n$$

3. The trace of \mathbf{A}_w is equal to the trace of \mathbf{A} :

$$\text{tr } \mathbf{A}_w = \text{tr } \mathbf{A} = z_1 + z_2 + \cdots + z_n$$

since the trace of a matrix, which is the sum of the diagonal elements, is equal to the sum of its characteristic values. The latter property follows immediately from the first property.

4. As is shown in Section 2.10,

$$\mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + \mathbf{D} = \mathbf{C}_w[z\mathbf{I} - \mathbf{A}_w]^{-1}\mathbf{B}_w + \mathbf{D}_w$$

Example 2.21

For the system of Example 2.20, one state-variable model given is

$$\begin{aligned}\mathbf{x}(k+1) &= \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k)\end{aligned}$$

We arbitrarily choose a linear-transformation matrix

$$\mathbf{P} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

The inverse of \mathbf{P} is given by

$$\mathbf{P}^{-1} = \frac{[\text{Cof}[\mathbf{P}]]^T}{|\mathbf{P}|}$$

where $[\cdot]^T$ indicates the transpose, and $\text{Cof}[\mathbf{P}]$ denotes the matrix of cofactors of \mathbf{P} . Thus

$$\text{Cof}[\mathbf{P}] = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

and

$$|\mathbf{P}| = 2$$

Then

$$\mathbf{P}^{-1} = \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix}$$

Thus, from (2-71),

$$\begin{aligned}\mathbf{A}_w &= \mathbf{P}^{-1} \mathbf{A} \mathbf{P} = \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1.8 & 0.2 \\ 0.9 & 0.9 \end{bmatrix} = \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} \\ \mathbf{B}_w &= \mathbf{P}^{-1} \mathbf{B} = \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \\ \mathbf{C}_w &= \mathbf{C} \mathbf{P} = [1 \quad 0] \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = [1 \quad -1]\end{aligned}$$

Then the new state equations are

$$\begin{aligned}\mathbf{w}(k+1) &= \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} \mathbf{w}(k) + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} u(k) \\ y(k) &= [1 \quad -1] \mathbf{w}(k)\end{aligned}$$

Note that

$$|z\mathbf{I} - \mathbf{A}| = \begin{vmatrix} z - 0.8 & -1 \\ 0 & z - 0.9 \end{vmatrix} = z^2 - 1.7z + 0.72 = (z - 0.8)(z - 0.9)$$

and

$$|z\mathbf{I} - \mathbf{A}_w| = \begin{vmatrix} z - 1.35 & -0.55 \\ 0.45 & z - 0.35 \end{vmatrix} = z^2 - 1.7z + 0.72$$

Hence the characteristic values are $z_1 = 0.8$ and $z_2 = 0.9$. Also,

$$|\mathbf{A}| = |\mathbf{A}_w| = 0.72 = z_1 z_2$$

and

$$\text{tr } \mathbf{A} = \text{tr } \mathbf{A}_w = 1.7 = z_1 + z_2$$

The results of this example can be verified with the MATLAB program CTRL, described in Appendix VII.

If a system has distinct characteristic values, we may derive a state-variable model in which the system matrix is diagonal. Consider a vector \mathbf{m}_i and a scalar z_i defined by the equation

$$\mathbf{A} \mathbf{m}_i = z_i \mathbf{m}_i \quad (2-76)$$

where $\mathbf{m}_i = [m_{1i} \quad m_{2i} \quad \cdots \quad m_{ni}]^T$. We express (2-76) as

$$(z_i \mathbf{I} - \mathbf{A}) \mathbf{m}_i = \mathbf{0}$$

For a nontrivial solution for this equation to exist, it is required that

$$|z_i \mathbf{I} - \mathbf{A}| = 0$$

Hence, from (2-72), it is seen that z_i is a characteristic value (also called an eigen-

value) of \mathbf{A} . The vector \mathbf{m}_i is called a *characteristic vector*, or *eigenvector*, of \mathbf{A} . From (2-76) we can construct the equation

$$\mathbf{A}[\mathbf{m}_1 \quad \mathbf{m}_2 \quad \cdots \quad \mathbf{m}_n] = [\mathbf{m}_1 \quad \mathbf{m}_2 \quad \cdots \quad \mathbf{m}_n] \begin{bmatrix} z_1 & 0 & \cdot & 0 \\ 0 & z_2 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & z_n \end{bmatrix}$$

or

$$\mathbf{A}\mathbf{M} = \mathbf{M}\mathbf{\Lambda}$$

where \mathbf{M} , called the *modal matrix*, is composed of the characteristic vectors as columns, and $\mathbf{\Lambda}$ is a diagonal matrix with the characteristic values of \mathbf{A} as the diagonal elements. The characteristic vectors are linearly independent provided the characteristic values are distinct. Hence we can write

$$\mathbf{\Lambda} = \mathbf{M}^{-1}\mathbf{A}\mathbf{M} \quad (2-77)$$

Then, in the general similarity transformation, \mathbf{P} is equal to \mathbf{M} , the modal matrix. An example will be given to illustrate this procedure.

Example 2.22

For the system of Example 2.21



$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

From Example 2.21, the characteristic values are $z_1 = 0.8$ and $z_2 = 0.9$. From (2-76),

$$\begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{21} \end{bmatrix} = 0.8 \begin{bmatrix} m_{11} \\ m_{21} \end{bmatrix}$$

Thus

$$0.8m_{11} + m_{21} = 0.8m_{11}$$

$$0.9m_{21} = 0.8m_{21}$$

Hence $m_{21} = 0$ and m_{11} is arbitrary. Let $m_{11} = 1$. Also,

$$\begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \begin{bmatrix} m_{12} \\ m_{22} \end{bmatrix} = 0.9 \begin{bmatrix} m_{12} \\ m_{22} \end{bmatrix}$$

Then m_{22} is seen to be arbitrary; let $m_{22} = 1$. Then $m_{12} = 10$. The modal matrix and its inverse are

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} = \begin{bmatrix} 1 & 10 \\ 0 & 1 \end{bmatrix}$$

and

$$\mathbf{M}^{-1} = \begin{bmatrix} 1 & -10 \\ 0 & 1 \end{bmatrix}$$

From (2-71) and (2-77),

$$\Lambda = \mathbf{M}^{-1} \mathbf{A} \mathbf{M} = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.9 \end{bmatrix}$$

$$\mathbf{B}_w = \mathbf{M}^{-1} \mathbf{B} = \begin{bmatrix} -10 \\ 1 \end{bmatrix}$$

$$\mathbf{C}_w = \mathbf{C} \mathbf{M} = [1 \quad 10]$$

Hence the new state model is given by

$$\mathbf{w}(k+1) = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.9 \end{bmatrix} \mathbf{w}(k) + \begin{bmatrix} -10 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 10] \mathbf{w}(k)$$

A MATLAB program that calculates the characteristic vectors and the characteristic values is given by

```
A = [0.8 1; 0 0.9];
[V,D] = eig(A)
```

```
result: V: 1 0.995    D: 0.8 0
          0 0.0995    . 0 0.9
```

The matrix \mathbf{V} gives the characteristic vectors as columns (each determined only within a constant factor), and \mathbf{D} gives the characteristic values on its diagonal.

2.10 TRANSFER FUNCTIONS

In the techniques described earlier in this text for obtaining a state-variable formulation, we derived the state model from the transfer function. In this section we demonstrate how one may obtain the transfer function, given the state model.

As a first approach, given the state equations of a discrete-time system, we can construct a simulation diagram. The transfer function can then be obtained from the simulation diagram, using Mason's gain formula.

Example 2.23

Consider the state model derived in Example 2.21.

$$\mathbf{x}(k+1) = \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad -1] \mathbf{x}(k)$$

A simulation diagram, constructed from these equations, is shown in Figure 2-15.

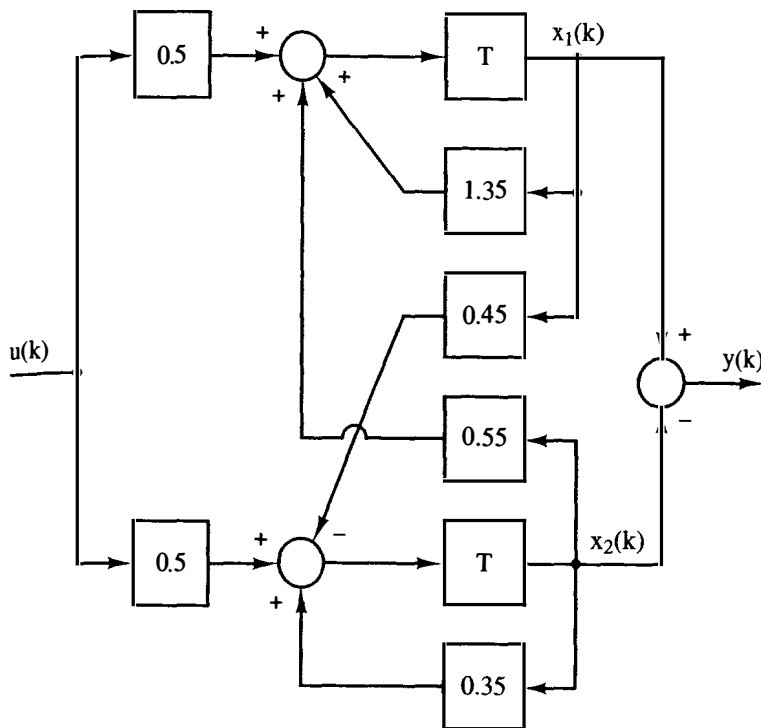


Figure 2-15 System for Example 2.23.

Application of Mason's gain formula to this figure yields

$$\begin{aligned} \frac{Y(z)}{U(z)} &= \frac{0.5z^{-1}(1 - 0.35z^{-1}) - 0.5z^{-1}(1 - 1.35z^{-1}) + (0.5)(0.45)z^{-2} + (0.5)(0.55)z^{-2}}{1 - 1.35z^{-1} - 0.35z^{-1} + (0.45)(0.55)z^{-2} + (1.35z^{-1})(0.35z^{-1})} \\ &= \frac{z^{-2}}{1 - 1.7z^{-1} + 0.72z^{-2}} = \frac{1}{z^2 - 1.7z + 0.72} \end{aligned}$$

Since this is the system transfer function, we see that the derived state model is valid.

Another approach for obtaining the system transfer function from the state equations is to take the z -transform of the state equations and eliminate $\mathbf{X}(z)$. Since

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (2-78)$$

taking the z -transform yields

$$z\mathbf{X}(z) - z\mathbf{x}(0) = \mathbf{A}\mathbf{X}(z) + \mathbf{B}U(z) \quad (2-79)$$

Since, in deriving transfer functions, we ignore initial conditions, (2-79) can be expressed as

$$[z\mathbf{I} - \mathbf{A}]\mathbf{X}(z) = \mathbf{B}U(z) \quad (2-80)$$

Solving for $\mathbf{X}(z)$, we obtain

$$\mathbf{X}(z) = [z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B}U(z) \quad (2-81)$$

Also, since

$$y(k) = \mathbf{C}\mathbf{x}(k) + Du(k) \quad (2-82)$$

then

$$Y(z) = \mathbf{C}\mathbf{X}(z) + DU(z) \quad (2-83)$$

Substituting (2-81) into (2-83) yields

$$Y(z) = [\mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + D]U(z)$$

The system transfer function is then seen to be

$$G(z) = \mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + D \quad (2-84)$$

This technique will now be illustrated by an example.

Example 2.24



Consider again the state equations of Example 2.23.

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} u(k) \\ y(k) &= [1 \quad -1] \mathbf{x}(k) \end{aligned}$$

Now

$$[z\mathbf{I} - \mathbf{A}] = \begin{bmatrix} z - 1.35 & -0.55 \\ 0.45 & z - 0.35 \end{bmatrix}$$

Thus

$$|z\mathbf{I} - \mathbf{A}| = z^2 - 1.7z + 0.72$$

Also,

$$\text{Cof}[z\mathbf{I} - \mathbf{A}] = \begin{bmatrix} z - 0.35 & -0.45 \\ 0.55 & z - 1.35 \end{bmatrix}$$

Then

$$[z\mathbf{I} - \mathbf{A}]^{-1} = \frac{[\text{Cof}[z\mathbf{I} - \mathbf{A}]]^T}{|z\mathbf{I} - \mathbf{A}|} = \frac{1}{z^2 - 1.7z + 0.72} \begin{bmatrix} z - 0.35 & 0.55 \\ -0.45 & z - 1.35 \end{bmatrix}$$

From (2-84), since $D = 0$,

$$\begin{aligned} G(z) &= \mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} \\ &= \frac{1}{z^2 - 1.7z + 0.72} [1 \quad -1] \begin{bmatrix} z - 0.35 & 0.55 \\ -0.45 & z - 1.35 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \\ &= \frac{1}{z^2 - 1.7z + 0.72} [1 \quad -1] \begin{bmatrix} 0.5z + 0.1 \\ 0.5z - 0.9 \end{bmatrix} \\ &= \frac{1}{z^2 - 1.7z + 0.72} \end{aligned}$$

This, of course, has been shown to be the transfer function.

Another property of the similarity transformation, applied to state models as given in (2-70), is seen from the example above. Since the transfer function of a system is invariant under a similarity transformation, then, from (2-70) and (2-84),

$$C[zI - A]^{-1}B + D = C_w[zI - A_w]^{-1}B_w + D_w \quad (2-85)$$

The proof of this property is left as an exercise for the reader (see Problem 2-35).

In this section two techniques are presented for obtaining the transfer function from the state equations. For high-order systems, the second method is more attractive. The second technique, as given in (2-84), can be implemented as a computer program [9]. For example, a MATLAB implementation of (2-84) for Example 2.24 is

```
A = [1.35  0.55;  -0.45  0.35];
B = [0.5;  0.5];
C = [1  -1];
D = 0;
[n,d] = ss2tf(A, B, C, D)
```

```
result:  n = 0.0    0.0  1.0
         d = 1.0   -1.7  0.72
```

2.11 SOLUTIONS OF THE STATE EQUATIONS

In this section we develop the general solution of linear time-invariant state equations. It will be seen that the key to the solution of the state equations is the calculation of the state transition matrix. Two related techniques, based on the z -transform, for calculating the state transition matrix are presented. Then the solution of state equations via the digital computer is mentioned.

Recursive Solution

We will first assume that the system is time invariant (fixed) and that $\mathbf{x}(0)$ and $\mathbf{u}(j), j = 0, 1, 2, \dots$, are known. Now the system equations are

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (2-86)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (2-87)$$

In a recursive manner it is obvious that

$$\mathbf{x}(1) = \mathbf{A}\mathbf{x}(0) + \mathbf{B}\mathbf{u}(0)$$

and

$$\mathbf{x}(2) = \mathbf{A}\mathbf{x}(1) + \mathbf{B}\mathbf{u}(1)$$

and hence

$$\begin{aligned}\mathbf{x}(2) &= \mathbf{A}(\mathbf{A}\mathbf{x}(0) + \mathbf{B}\mathbf{u}(0)) + \mathbf{B}\mathbf{u}(1) \\ &= \mathbf{A}^2\mathbf{x}(0) + \mathbf{A}\mathbf{B}\mathbf{u}(0) + \mathbf{B}\mathbf{u}(1)\end{aligned}$$

In a similar manner we can show that

$$\mathbf{x}(3) = \mathbf{A}^3\mathbf{x}(0) + \mathbf{A}^2\mathbf{B}\mathbf{u}(0) + \mathbf{A}\mathbf{B}\mathbf{u}(1) + \mathbf{B}\mathbf{u}(2)$$

It is seen, then, that the general solution is given by

$$\mathbf{x}(k) = \mathbf{A}^k\mathbf{x}(0) + \sum_{j=0}^{k-1} \mathbf{A}^{(k-1-j)}\mathbf{B}\mathbf{u}(j) \quad (2-88)$$

If we define

$$\Phi(k) = \mathbf{A}^k$$

then

$$\mathbf{x}(k) = \Phi(k)\mathbf{x}(0) + \sum_{j=0}^{k-1} \Phi(k-1-j)\mathbf{B}\mathbf{u}(j) \quad (2-89)$$

This equation is the general solution to (2-86). From (2-87) and (2-89),

$$\mathbf{y}(k) = \mathbf{C}\Phi(k)\mathbf{x}(0) + \sum_{j=0}^{k-1} \mathbf{C}\Phi(k-1-j)\mathbf{B}\mathbf{u}(j) + \mathbf{D}\mathbf{u}(k) \quad (2-90)$$

$\Phi(k)$ is called the *state transition matrix* or the *fundamental matrix*. An example will now be given to illustrate the recursive nature of the solution.

Example 2.25

Consider the transfer function

$$G(z) = \frac{(z+3)}{(z+1)(z+2)}$$

Using the technique of Section 2.8, we write

$$\begin{aligned}\mathbf{x}(k+1) &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [3 \quad 1] \mathbf{x}(k)\end{aligned}$$

Assume that the system is initially at rest so that $\mathbf{x}(0) = \mathbf{0}$, and that the input is a unit step; that is,

$$u(k) = 1, \quad k = 0, 1, 2, \dots$$

The recursive solution is obtained as follows:

$$\begin{aligned}\mathbf{x}(1) &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathbf{x}(0) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ y(1) &= [3 \quad 1] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1\end{aligned}$$



Then

$$\begin{aligned}\mathbf{x}(2) &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathbf{x}(1) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(1) = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (1) \\ &= \begin{bmatrix} 1 \\ -2 \end{bmatrix} \\ y(2) &= [3 \quad 1] \begin{bmatrix} 1 \\ -2 \end{bmatrix} = 3 - 2 = 1\end{aligned}$$

In a similar manner it can be shown that

$$\mathbf{x}(3) = \begin{bmatrix} -2 \\ 5 \end{bmatrix}, \quad y(3) = -1$$

and

$$\mathbf{x}(4) = \begin{bmatrix} 5 \\ -10 \end{bmatrix}, \quad y(4) = 5, \text{ etc.}$$

Hence one can recursively determine the states and the output at successive time instants.

A MATLAB program for this example is given by

```
A = [0 1;-2 -3];
B = [0;1];
C = [3 1];
x = [0;0];
u=1;
for k = 0:5
    x1 = A*x + B*u;
    y = C*x
    x = x1;
end

result: y = 1  1  -1  5  -9
```

z-Transform Method

The general solution to the state equations

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (2-91)$$

was developed above, and is given by

$$[\text{eq. (2-89)}] \quad \mathbf{x}(k) = \Phi(k)\mathbf{x}(0) + \sum_{j=0}^{k-1} \Phi(k-1-j)\mathbf{B}u(j)$$

where $\Phi(k)$, the state transition matrix, is given by

$$\Phi(k) = \mathbf{A}^k \quad (2-92)$$

One technique for evaluating $\Phi(k)$ as a function of k is through the use of the z-transform. This technique will now be presented.

In (2-91), let $\mathbf{u}(k) = \mathbf{0}$. Then the z-transform of this equation yields

$$z\mathbf{X}(z) - z\mathbf{x}(0) = \mathbf{A}\mathbf{X}(z) \quad (2-93)$$

Solving for $\mathbf{X}(z)$, we see that

$$\mathbf{X}(z) = z[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{x}(0) \quad (2-94)$$

Then

$$\mathbf{x}(k) = \mathcal{Z}^{-1}[\mathbf{X}(z)] = \mathcal{Z}^{-1}[z[z\mathbf{I} - \mathbf{A}]^{-1}]\mathbf{x}(0) \quad (2-95)$$

Comparing (2-95) with (2-89), we see that

$$\Phi(k) = \mathcal{Z}^{-1}[z[z\mathbf{I} - \mathbf{A}]^{-1}] \quad (2-96)$$

To illustrate this method, consider the following example.

Example 2.26

For the state equations of Example 2.25,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

Then

$$[z\mathbf{I} - \mathbf{A}] = \begin{bmatrix} z & -1 \\ 2 & z + 3 \end{bmatrix}$$

and

$$|z\mathbf{I} - \mathbf{A}| = z^2 + 3z + 2 = (z + 1)(z + 2)$$

Evaluating the inverse matrix in (2-96) and multiplying by z , we obtain

$$\begin{aligned} z[z\mathbf{I} - \mathbf{A}]^{-1} &= \begin{bmatrix} \frac{z(z+3)}{(z+1)(z+2)} & \frac{z}{(z+1)(z+2)} \\ \frac{-2z}{(z+1)(z+2)} & \frac{z^2}{(z+1)(z+2)} \end{bmatrix} \\ &= \begin{bmatrix} \frac{2z}{z+1} + \frac{-z}{z+2} & \frac{z}{z+1} + \frac{-z}{z+2} \\ \frac{-2z}{z+1} + \frac{2z}{z+2} & \frac{-z}{z+1} + \frac{2z}{z+2} \end{bmatrix} = \mathcal{Z}(\Phi(k)) \end{aligned}$$

Thus

$$\Phi(k) = \mathcal{Z}^{-1}[z[z\mathbf{I} - \mathbf{A}]^{-1}] = \begin{bmatrix} 2(-1)^k - (-2)^k & (-1)^k - (-2)^k \\ -2(-1)^k + 2(-2)^k & -(-1)^k + 2(-2)^k \end{bmatrix}$$

Numerical Method via Digital Computer

The digital computer is ideally suited for evaluating equations of the type

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (2-97)$$

These equations can be solved recursively by the computer without actually solving for the state transition matrix.

Another computer method for finding the state transition matrix is to evaluate the expression

$$\Phi(k) = A^k \quad (2-98)$$

The disadvantage of this procedure is that $\Phi(k)$ is not found as a general function of k . However, for high-order systems, the evaluation of $\Phi(k)$ as a function of k is difficult using any method.

Properties of the State Transition Matrix

Three properties of the state transition matrix will now be derived. Since

$$\mathbf{x}(k) = \Phi(k)\mathbf{x}(0) \quad (2-99)$$

then evaluating this expression for $k = 0$ yields the first property:

$$\Phi(0) = \mathbf{I} \quad (2-100)$$

where \mathbf{I} is the identity matrix. Next, since

$$\Phi(k) = A^k$$

then the second property is given by

$$\Phi(k_1 + k_2) = A^{k_1 + k_2} = A^{k_1} A^{k_2} = \Phi(k_1)\Phi(k_2) \quad (2-101)$$

The third property is seen from the relationships

$$\Phi(-k) = A^{-k} = [A^k]^{-1} = \Phi^{-1}(k) \quad (2-102)$$

or, taking the inverse of this expression, we obtain an equivalent expression

$$\Phi(k) = \Phi^{-1}(-k) \quad (2-103)$$

Example 2.27

We will illustrate the properties of the state transition matrix using the system described by

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}(k)$$

Now

$$\Phi(k) = A^k = \begin{bmatrix} 1^k & 0 \\ 0 & 0.5^k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^k \end{bmatrix}$$

Then, in (2-100),

$$\Phi(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}$$

In (2-101),

$$\Phi(k_1 + k_2) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^{k_1 + k_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^{k_1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0.5^{k_2} \end{bmatrix} = \Phi(k_1)\Phi(k_2)$$

Also, in (2-103),

$$\Phi^{-1}(-k) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^{-k} \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^k \end{bmatrix} = \Phi(k)$$

2.12 LINEAR TIME-VARYING SYSTEMS

The state equations for a linear time-varying discrete system were given in (2-47) and (2-48), and are repeated here.

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (2-104)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) \quad (2-105)$$

We can find the solution to these equations in a recursive manner. If we denote initial time by k_0 and assume that $\mathbf{x}(k_0)$ is known and that $\mathbf{u}(k)$ is known for $k \geq k_0$, then

$$\mathbf{x}(1 + k_0) = \mathbf{A}(k_0)\mathbf{x}(k_0) + \mathbf{B}(k_0)\mathbf{u}(k_0)$$

Thus

$$\begin{aligned} \mathbf{x}(2 + k_0) &= \mathbf{A}(1 + k_0)\mathbf{x}(1 + k_0) + \mathbf{B}(1 + k_0)\mathbf{u}(1 + k_0) \\ &= \mathbf{A}(1 + k_0)[\mathbf{A}(k_0)\mathbf{x}(k_0) + \mathbf{B}(k_0)\mathbf{u}(k_0)] + \mathbf{B}(1 + k_0)\mathbf{u}(1 + k_0) \\ &= \mathbf{A}(1 + k_0)\mathbf{A}(k_0)\mathbf{x}(k_0) + \mathbf{A}(1 + k_0)\mathbf{B}(k_0)\mathbf{u}(k_0) + \mathbf{B}(1 + k_0)\mathbf{u}(1 + k_0) \end{aligned}$$

In a like manner, we see that

$$\begin{aligned} \mathbf{x}(3 + k_0) &= \mathbf{A}(2 + k_0)\mathbf{x}(2 + k_0) + \mathbf{B}(2 + k_0)\mathbf{u}(2 + k_0) \\ &= \mathbf{A}(2 + k_0)\mathbf{A}(1 + k_0)\mathbf{A}(k_0)\mathbf{x}(k_0) + \mathbf{A}(2 + k_0)\mathbf{A}(1 + k_0)\mathbf{B}(k_0)\mathbf{u}(k_0) \\ &\quad + \mathbf{A}(2 + k_0)\mathbf{B}(1 + k_0)\mathbf{u}(1 + k_0) + \mathbf{B}(2 + k_0)\mathbf{u}(2 + k_0) \end{aligned}$$

Now, if we define

$$\Phi(k, k_0) = \mathbf{A}(k-1)\mathbf{A}(k-2)\cdots\mathbf{A}(k_0) = \begin{cases} \prod_{j=k_0}^{k-1} \mathbf{A}(j), & k > k_0 \\ \mathbf{I}, & k = k_0 \end{cases} \quad (2-106)$$

where \mathbf{I} is the identity matrix, then the equation above for $\mathbf{x}(3 + k_0)$ can be written as

$$\mathbf{x}(3 + k_0) = \Phi(3 + k_0, k_0)\mathbf{x}(k_0) + \sum_{j=k_0}^{2+k_0} \Phi(3 + k_0, j+1)\mathbf{B}(j)\mathbf{u}(j)$$

Thus it can be shown in general that

$$\mathbf{x}(k) = \Phi(k, k_0)\mathbf{x}(k_0) + \sum_{j=k_0}^{k-1} \Phi(k, j+1)\mathbf{B}(j)\mathbf{u}(j) \quad (2-107)$$

and then

$$y(k) = C(k)\Phi(k, k_0)x(k_0) + \sum_{j=k_0}^{k-1} C(k)\Phi(k, j+1)B(j)u(j) + D(k)u(k) \quad (2-108)$$

As before, $\Phi(k, k_0)$ is called the state transition matrix. Note that since the matrices are time varying, they must be reevaluated at each time instant.

Using (2-106) and (2-107), we can derive the following important properties of the state transition matrix $\Phi(k, k_0)$:

$$\begin{aligned} \Phi(k_0, k_0) &= I \\ \Phi(k_2, k_1)\Phi(k_1, k_0) &= \Phi(k_2, k_0) \\ \Phi(k_1, k_2) &= \Phi^{-1}(k_2, k_1) \end{aligned} \quad (2-109)$$

Note that if the system is time invariant, then A is not a function k . The equation for the state transition matrix, (2-106), then reduces to that for the time-invariant system, (2-92). It is seen that the time-invariant system is a special case of the time-varying system.

2.13 SUMMARY

In this chapter we have introduced the concepts of discrete-time systems and the modeling of these systems by difference equations. The z -transform was defined and was shown to be applicable to the solution of linear time-invariant difference equations. Next, four methods for determining inverse z -transforms were presented. Finally, the representation of discrete-time systems by simulation diagrams and flow graphs was introduced, which leads naturally to state-variable modeling. Techniques for the solution of linear state equations were then presented. The foundations for the modeling and the analysis of discrete-time systems were presented in this chapter, and will serve for much of the mathematical basis for the following chapters.

REFERENCES AND FURTHER READING

1. C. F. Gerald, *Applied Numerical Analysis*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1984.
2. "Software Implementation ALS Computer Program," Contract N00421-75-C-0058, Bell Aerospace Corporation, Buffalo, NY, Mar. 1975.
3. A. W. Drake, *Fundamentals of Applied Probability Theory*. New York: McGraw-Hill Book Company, 1967.
4. A. N. Oppenheim and A. S. Willsky, *Signals and Systems*. Englewood Cliffs, NJ: Prentice Hall, 1983.
5. G. Doetsch, *Guide to the Applications of the Laplace and z -Transforms*. New York: Van Nostrand Reinhold, 1971.

6. M. M. Guterman and Z. H. Nitecki, *Differential Equations: A First Course*, 3d ed. New York: Saunders College Publishing, 1991.
7. C. R. Wylie, Jr., *Advanced Engineering Mathematics*, 4th ed. New York: McGraw-Hill Book Company, 1975.
8. P. M. De Russo, R. J. Roy, and C. M. Close, *State Variables for Engineers*. New York: John Wiley & Sons, Inc., 1965.
9. B. Friedland, *Control System Design*. New York: McGraw-Hill Book Company, 1986.
10. J. A. Cadzow and H. R. Martens, *Digital-Time and Computer Control Systems*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1970.
11. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1988.
12. E. I. Jury, *Theory and Application of the z-Transform Method*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.
13. B. C. Kuo, *Digital Control Systems*, 2d ed. New York: Saunders College Publishing, 1992.

PROBLEMS

- 2-1. Find the z-transform of the number sequence generated by sampling the time function $e(t) = t$ every T seconds, beginning at $t = 0$. Can you express this transform in closed form?
- 2-2. (a) Write, as a series, the z-transform of the number sequence generated by sampling the time function $e(t) = e^{-t}$ every T seconds, beginning at $t = 0$. Can you express this transform in closed form?
 (b) Evaluate the coefficients in the series of part (a) for the case that $T = 0.05$ s.
 (c) The exponential $e(t) = e^{-bt}$ is sampled every $T = 0.2$ s, yielding the z-transform

$$E(z) = 1 + \left(\frac{1}{2}\right)z^{-1} + \left(\frac{1}{2}\right)^2 z^{-2} + \left(\frac{1}{2}\right)^3 z^{-3} + \dots$$

Evaluate b .

- 2-3. Find the z-transforms of the number sequences generated by sampling the following time functions every T seconds, beginning at $t = 0$. Express these transforms in closed form.
 (a) $e(t) = e^{-at}$
 (b) $e(t) = e^{-(t-T)}u(t-T)$
 (c) $e(t) = e^{-(t-5T)}u(t-5T)$
- 2-4. Find the z-transform, in closed form, of the number sequence generated by sampling the time function $e(t)$ every T seconds beginning at $t = 0$. The function $e(t)$ is specified by its Laplace transform,

$$E(s) = \frac{2(1 - e^{-5s})}{s(s+2)}, \quad T = 1 \text{ s}$$

- 2-5. (a) Find $e(0)$, $e(1)$, and $e(10)$ for

$$E(z) = \frac{0.1}{z(z - 0.9)}$$

using the inversion formula.

- (b) Check the value of $e(0)$ using the initial-value property.
- (c) Check the values calculated in part (a) using partial fractions.
- (d) Find $e(k)$ for $k = 0, 1, 2, 3$, and 4 if $\mathcal{Z}[e(k)]$ is given by

$$E(z) = \frac{1.98z}{(z^2 - 0.9z + 0.9)(z - 0.8)(z^2 - 1.2z + 0.27)}$$

- (e) Find a function $e(t)$ which, when sampled at a rate of 10 Hz ($T = 0.1$ s), results in the transform $E(z) = 2z/(z - 0.8)$.
- (f) Repeat part (e) for $E(z) = 2z/(z + 0.8)$.
- (g) From parts (e) and (f), what is the effect on the inverse z -transform of changing the sign on a real pole?

2-6. A function $e(t)$ is sampled, and the resultant sequence has the z -transform

$$E(z) = \frac{z^3 - 2z}{z^4 - 0.9z^2 + 0.8}$$

Solve this problem using $E(z)$ and the properties of the z -transform.

- (a) Find the z -transform of $e(t - 2T)u(t - 2T)$.
- (b) Find the z -transform of $e(t + 2T)u(t)$.
- (c) Find the z -transform of $e(t - T)u(t - 2T)$.

2-7. For the number sequence $\{e(k)\}$,

$$E(z) = \frac{z}{(z + 1)^2}$$

- (a) Apply the final-value theorem to $E(z)$.
- (b) Check your result in part (a) by finding the inverse z -transform of $E(z)$.
- (c) Repeat parts (a) and (b) with $E(z) = z/(z - 1)^2$.
- (d) Repeat parts (a) and (b) with $E(z) = z/(z - 0.9)^2$.
- (e) Repeat parts (a) and (b) with $E(z) = z/(z - 1.1)^2$.

2-8. Find the inverse z -transform of each $E(z)$ below by the four methods given in the text. Compare the values of $e(k)$, for $k = 0, 1, 2, 3$, obtained by the four methods.

$$(a) E(z) = \frac{0.5z}{(z - 1)(z - 0.6)}$$

$$(b) E(z) = \frac{0.5}{(z - 1)(z - 0.6)}$$

$$(c) E(z) = \frac{0.5(z + 1)}{(z - 1)(z - 0.6)}$$

$$(d) E(z) = \frac{z(z - 0.7)}{(z - 1)(z - 0.6)}$$

- (e) Use MATLAB to verify the partial-fraction expansions.

2-9. From Table 2-3,

$$\mathcal{Z}[\cos ak] = \frac{z(z - \cos a)}{z^2 - 2z \cos a + 1}$$

- (a) Find the conditions on the parameter a such that $\mathcal{Z}[\cos ak]$ is first order (pole-zero cancellation occurs).
- (b) Give the first-order transfer function in part (a).
- (c) Find a such that $\mathcal{Z}[\cos ak] = \mathcal{Z}[u(k)]$, where $u(k)$ is the unit-step function.

2-10. Solve the given difference equation for $x(k)$ using:

- (a) The sequential technique.

(b) The z-transform.

(c) Will the final-value theorem give the correct value of $x(k)$ as $k \rightarrow \infty$?

$$x(k) - 3x(k-1) + 2x(k-2) = e(k)$$

where

$$e(k) = \begin{cases} 1, & k = 0, 1 \\ 0, & k \geq 2 \end{cases}$$

$$x(-2) = x(-1) = 0$$

2-11. Given the difference equation

$$y(k+2) - \frac{3}{4}y(k+1) + \frac{1}{8}y(k) = e(k)$$

where $y(0) = y(1) = 0$, $e(0) = 0$, and $e(k) = 1$, $k = 1, 2, \dots$

(a) Solve for $y(k)$ as a function of k . Give the numerical values of $y(k)$, $0 \leq k \leq 4$.

(b) Solve the difference equation directly for $y(k)$, $0 \leq k \leq 4$, to verify the results of part (a).

(c) Repeat parts (a) and (b) for $e(k) = 0$ for all k , and $y(0) = 1$, $y(1) = -2$.

2-12. Given the difference equation

$$x(k) - x(k-1) + x(k-2) = e(k)$$

where $e(k) = 1$ for $k \geq 0$.

(a) Solve for $x(k)$ as a function of k , using the z-transform. Give the values of $x(0)$, $x(1)$, and $x(2)$.

(b) Verify the values $x(0)$, $x(1)$, and $x(2)$, using the power-series method.

(c) Verify the values $x(0)$, $x(1)$, and $x(2)$ by solving the difference equation directly.

(d) Will the final-value property give the correct value for $x(\infty)$?

2-13. Given the difference equation

$$x(k+2) + 3x(k+1) + 2x(k) = e(k)$$

where

$$e(k) = \begin{cases} 1, & k = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$x(0) = 1$$

$$x(1) = -1$$

(a) Solve for $x(k)$ as a function of k .

(b) Evaluate $x(0)$, $x(1)$, $x(2)$, and $x(3)$ in part (a).

(c) Verify the results in part (b) using the power-series method.

(d) Verify the results in part (b) by solving the difference equation directly.

2-14. Given the difference equation

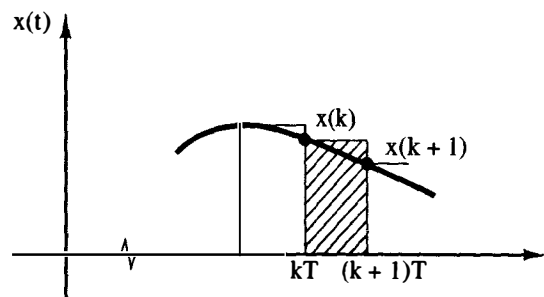
$$x(k+3) - 2.2x(k+2) + 1.57x(k+1) - 0.36x(k) = e(k).$$

where $e(k) = 1$ for all $k \geq 0$, and $x(0) = x(1) = x(2) = 0$.

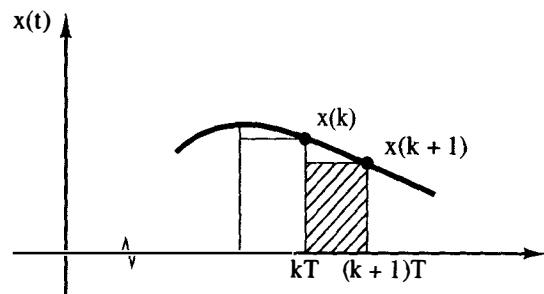
(a) Write a digital computer program that will calculate $x(k)$. Run this program solving for $x(3)$, $x(4)$, \dots , $x(25)$.

(b) Using the sequential technique, check the values of $x(k)$, $0 \leq k \leq 5$.

- (c) Use the z -transform and the power-series method to verify the values $x(k)$, $0 \leq k \leq 5$.
- 2-15.** The rectangular rules for numerical integration are illustrated in Figure P2-15. The left-side rule is depicted in Figure P2-15a, and the right-side rule is depicted in Figure P2-15b. The integral of $x(t)$ is approximated by the sum of the rectangular areas shown for each rule. Let $y(kT)$ be the numerical integral of $x(t)$, $0 \leq t \leq kT$.
- Write the difference equation relating $y(k+1)$, $y(k)$, and $x(k)$ for the left-side rule.
 - Find the transfer function $Y(z)/X(z)$ for part (a).
 - Write the difference equation relating $y(k+1)$, $y(k)$, and $x(k+1)$ for the right-side rule.
 - Find the transfer function $Y(z)/X(z)$ for part (c).
 - Express $y(k)$ as a summation on $x(k)$ for the left-side rule.
 - Express $y(k)$ as a summation on $x(k)$ for the right-side rule.



(a)



(b)

Figure P2-15 Rectangular rules for integration: (a) left side; (b) right side.

- 2-16.** The trapezoidal rule (modified Euler method) for numerical integration approximates the integral of a function $x(t)$ by summing trapezoid areas as shown in Figure P2-16. Let $y(t)$ be the integral of $x(t)$.
- Write the difference equation relating $y[(k+1)T]$, $y(kT)$, $x[(k+1)T]$, and $x(kT)$ for this rule.

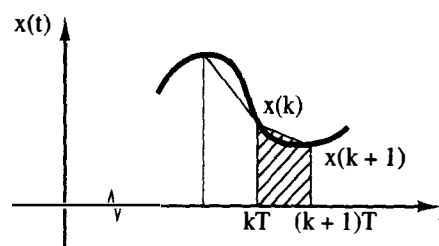


Figure P2-16 Trapezoidal rule for numerical integration.

- (b) Show that the transfer function for this integrator is given by

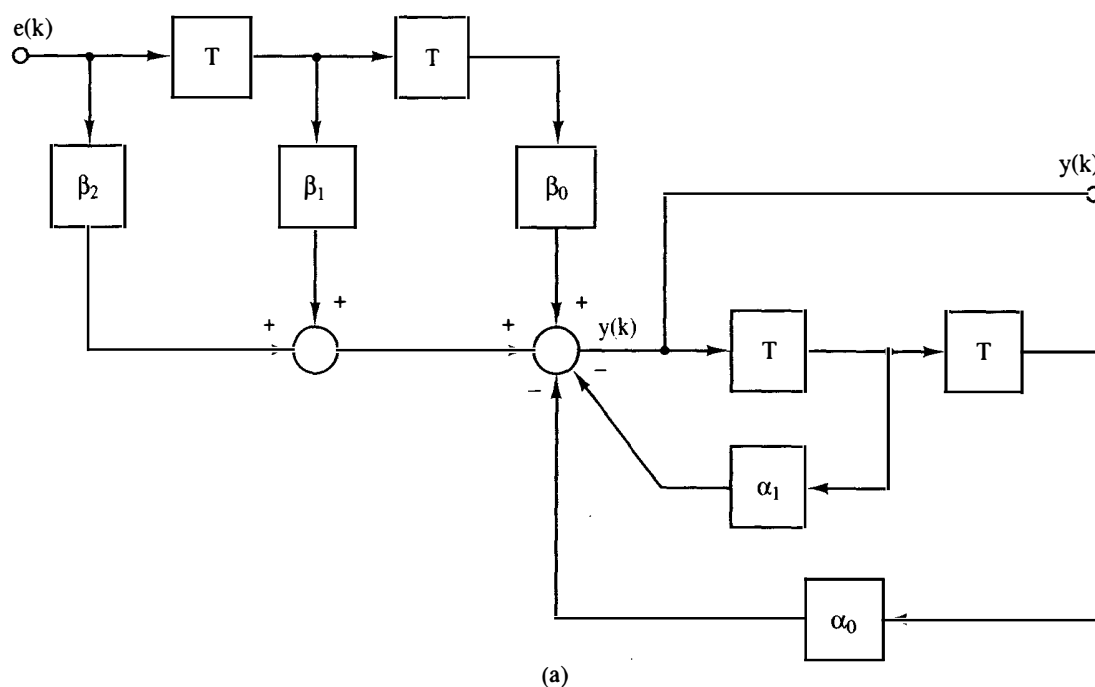
$$\frac{Y(z)}{X(z)} = \frac{(T/2)(z + 1)}{z - 1}$$

- 2-17. (a) The transfer function for the right-side rectangular-rule integrator was found in Problem 2-15 to be $Y(z)/X(z) = Tz/(z - 1)$. We would suspect that the reciprocal of this transfer function should yield an approximation to a differentiator. That is, if $w(kT)$ is a numerical derivative of $x(t)$ at $t = kT$,

$$\frac{W(z)}{X(z)} = \frac{z - 1}{Tz}$$

Write the difference equation describing this differentiator.

- (b) Draw a figure similar to those in Figure P2-15 illustrating the approximate differentiation.
- (c) Repeat part (a) for the left-side rule, where $W(z)/X(z) = T/(z - 1)$.
- (d) Repeat part (b) for the differentiator of part (c).
- 2-18. Given in Figure P2-18 are two digital-filter structures, or realizations, for second-order filters. These two structures, along with several additional ones, are discussed in Chapter 12.
- (a) Write the difference equation for the 3D structure of Figure P2-18a, expressing $y(k)$ as a function of $y(k - i)$ and $e(k - i)$.
- (b) Derive the filter transfer function $Y(z)/E(z)$ for the 3D structure by taking the z-transform of the equation in part (a).
- (c) Write the difference equation for the 1D structure of Figure P2-18b. Two equations are required, with one for $f(k)$ and one for $y(k)$.



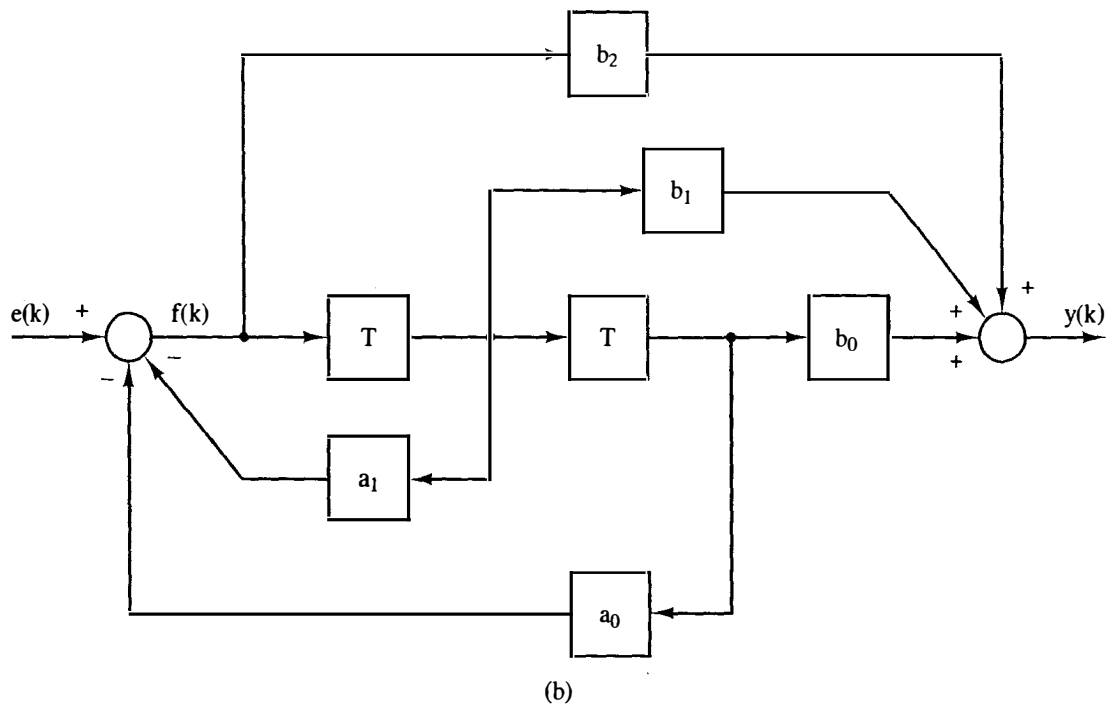


Figure P2-18 Digital filter structures: (a) 3D; (b) 1D.

- (d) Derive the filter transfer function $Y(z)/E(z)$ for the 1D structure by taking the z -transform of the equations in part (c), and eliminating $F(z)$.
- (e) From parts (b) and (d), relate the coefficients α_i, β_i to a_i, b_i such that the two filters realize the same transfer function.
- (f) Write a computer-program segment that realizes the 3D structure. This program should be of the form of that of Section 2.5.
- (g) Write a computer-program segment that realizes the 1D structure. This program should be of the form of that of Section 2.5.
- 2-19. Shown in Figure P2-19 is the second-order digital-filter structure 1X. This structure realizes the filter transfer function

$$D(z) = b_2 + \frac{A}{z - p} + \frac{A^*}{z - p^*}$$

where p and p^* (conjugate of p) are complex. The relationships between the filter coefficients and the coefficients in Figure P2-19 are given by

$$\begin{aligned} g_1 &= \operatorname{Re}(p) & g_3 &= -2 \operatorname{Im}(A) \\ g_2 &= \operatorname{Im}(p) & g_4 &= 2 \operatorname{Re}(A) \end{aligned}$$

- (a) To realize this filter, difference equations are required for $f_1(k)$, $f_2(k)$, and $y(k)$. Write these equations.
- (b) Find the filter transfer function $Y(z)/E(z)$ by taking the z -transform of the equations of part (a) and eliminating $F_1(z)$ and $F_2(z)$.
- (c) Verify the results in part (b) using Mason's gain formula.
- (d) Write a computer-program segment that realizes the 1X structure. This program should be of the form of that of Section 2.5.

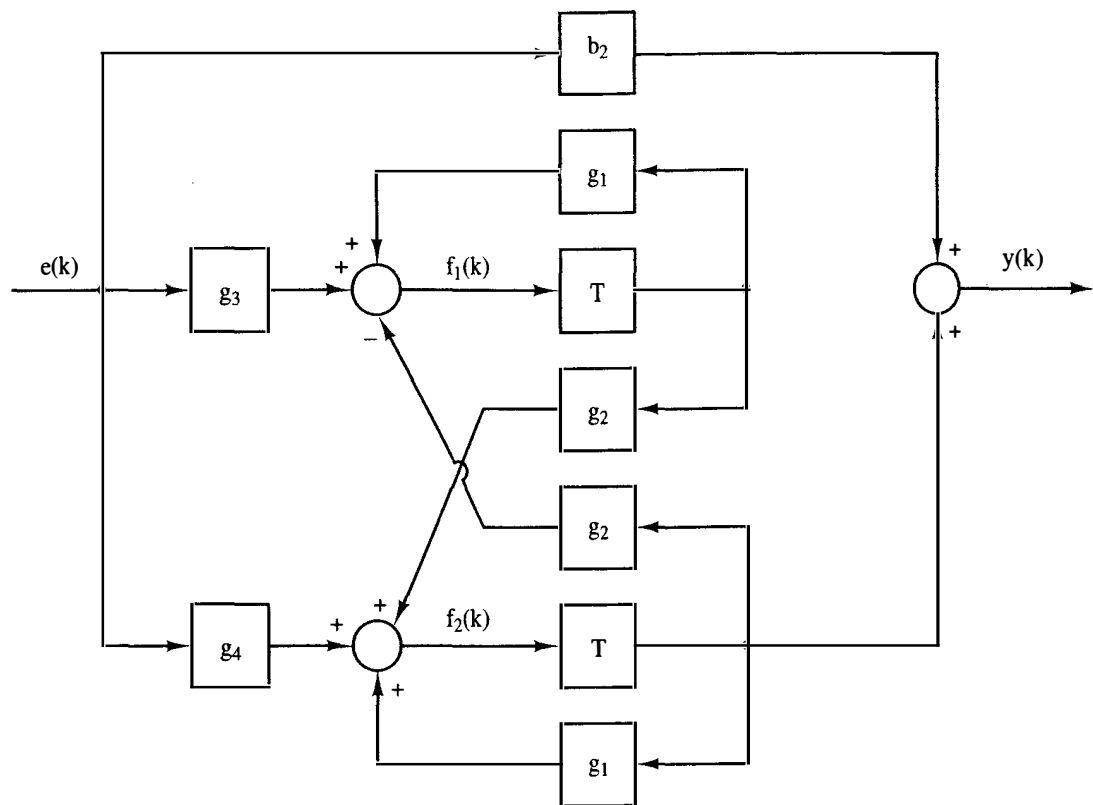


Figure P2-19 Digital-filter structure 1X.

2-20. Given the second-order digital-filter transfer function

$$D(z) = \frac{2z^2 - 2.4z + 0.72}{z^2 - 1.4z + 0.98}$$

- Find the coefficients of the 3D structure of Figure P2-18 such that $D(z)$ is realized.
- Find the coefficients of the 1D structure of Figure P2-18 such that $D(z)$ is realized.
- Find the coefficients of the 1X structure of Figure P2-19 such that $D(z)$ is realized. The coefficients are identified in Problem 2-19.
- Use MATLAB to verify the partial-fraction expansions in part (c).
- Verify the results in part (c) using Mason's gain formula.

2-21. Given the MATLAB program that solves the difference equation of a digital controller.



```
s1 = 0;
e = 0;
for k = 0:5
    s2 = e - s1;
    m = 0.5*s2 - s1;
    s1 = s2;
    [k,m]
    e = e + 1;
end
```

- Find the transfer function of the controller.

- (b) Find the z -transform of the controller input.
 - (c) Use the results of parts (a) and (b) to find the inverse z -transform of the controller output.
 - (d) Run the program to check the results of part (c).
- 2-22.** Find two different state-variable formulations that model the system whose difference equation is given by:
- (a) $y(k+2) + 6y(k+1) + 5y(k) = 2e(k)$
 - (b) $y(k+2) + 6y(k+1) + 5y(k) = e(k+1) + 2e(k)$
 - (c) $y(k+2) + 6y(k+1) + 5y(k) = 3e(k+2) + e(k+1) + 2e(k)$
- 2-23.** Consider a system with the transfer function

$$G(z) = \frac{Y(z)}{U(z)} = \frac{2}{z(z-1)}$$

- (a) Find three different state-variable models of this system.
 - (b) Verify the transfer function of each state model in part (a), using (2-84).
 - (c) Use MATLAB to verify each transfer function.
- 2-24.** Consider a system described by the coupled difference equation

$$\begin{aligned} y(k+2) - v(k) &= 0 \\ v(k+1) + y(k+1) &= u(k) \end{aligned}$$

where $u(k)$ is the system input.

- (a) Find a state-variable formulation for this system. Consider the outputs to be $y(k+1)$ and $v(k)$. *Hint:* Draw a simulation diagram first.
 - (b) Repeat part (a) with $y(k)$ and $v(k)$ as the outputs.
 - (c) Repeat part (a) with the single output $v(k)$.
 - (d) Use (2-84) to calculate the system transfer function with $v(k)$ as the system output, as in part (c); that is, find $V(z)/U(z)$.
 - (e) Verify the transfer function $V(z)/U(z)$ in part (d) by taking the z -transform of the given system difference equations and eliminating $Y(z)$.
 - (f) Verify the transfer function $V(z)/U(z)$ in part (d) by using Mason's gain formula on the simulation diagram of part (a).
 - (g) Verify the transfer function using MATLAB.
- 2-25.** Find a state-variable formulation for the system described by the coupled second-order difference equations given. The system output is $y(k)$, and $e_1(k)$ and $e_2(k)$ are the system inputs. *Hint:* Draw a simulation diagram first.

$$\begin{aligned} x(k+2) + v(k+1) &= 4e_1(k) + e_2(k) \\ v(k+2) - v(k) + x(k) &= 2e_1(k) \\ y(k) &= v(k+2) - x(k+1) + e_1(k) \end{aligned}$$

- 2-26.** Consider the system described by

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 0 & 1 \\ 0 & 3 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [-2 \quad 1] \mathbf{x}(k) \end{aligned}$$

- (a) Find the transfer function $Y(z)/U(z)$.

- (b) Using any similarity transformation, find a different state model for this system.
- (c) Find the transfer function of the system from the transformed state equations.
- (d) Verify that \mathbf{A} given and \mathbf{A}_w derived in part (b) satisfy the first three properties of similarity transformations. The fourth property was verified in part (c).
- (e) Verify the results in part (c) using MATLAB.

2-27. Consider the system of Problem 2-26. A similarity transformation on these equations yields

$$\mathbf{w}(k+1) = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \mathbf{w}(k) + \mathbf{B}_w u(k)$$

$$y(k) = \mathbf{C}_w \mathbf{x}(k)$$

- (a) Find d_1 and d_2 .
 - (b) Find a similarity transformation that results in the \mathbf{A}_w matrix given. Note that this matrix is diagonal.
 - (c) Find \mathbf{B}_w and \mathbf{C}_w .
 - (d) Find the transfer functions of both sets of state equations to verify the results of this problem.
 - (e) Verify the results in part (d) using MATLAB.
- 2-28. Consider the system described in Problem 2-26.
- (a) Find the transfer function of this system.
 - (b) Let $u(k) = 1, k \geq 0$ (a unit step function) and $\mathbf{x}(0) = \mathbf{0}$. Use the transfer function of part (a) to find the system response.
 - (c) Find the state transition matrix $\Phi(k)$ for this system.
 - (d) Use (2-90) to verify the step response calculated in part (b). This calculation results in the response expressed as a summation. Then check the values $y(0)$, $y(1)$, and $y(2)$.
 - (e) Verify the results of part (d) by the iterative solution of the state equations.

2-29. Repeat Problem 2-26 for the system described by

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 2] \mathbf{x}(k)$$

2-30. The system described by the equations

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 2] \mathbf{x}(k)$$

is excited by the initial conditions $\mathbf{x}(0) = [1 \quad 2]^T$ with $u(k) = 0$ for all k .

- (a) Use (2-89) to solve for $\mathbf{x}(k), k \geq 0$.
- (b) Find the output $y(k)$.
- (c) Show that $\Phi(k)$ in (a) satisfies the property $\Phi(0) = \mathbf{I}$.
- (d) Show that the solution in part (a) satisfies the given initial conditions.
- (e) Use an iterative solution of the state equations to show that the values $y(k)$, for $k = 0, 1, 2$, and 3 , in part (b) are correct.
- (f) Verify the results in part (e) using MATLAB.

2-31. The system described by the equations

$$\mathbf{x}(k+1) = \begin{bmatrix} 1.1 & 1 \\ -0.3 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad -1] \mathbf{x}(k)$$

is excited by the initial conditions $\mathbf{x}(0) = [-1 \quad 2]^T$ with $u(k) = 0$ for all k .

- Use (2-89) to solve for $\mathbf{x}(k)$, $k \geq 0$.
- Find the output $y(k)$.
- Show that $\Phi(k)$ in part (a) satisfies the property $\Phi(0) = \mathbf{I}$.
- Show that the solution in part (a) satisfies the given initial conditions.
- Use an iterative solution of the state equations to show that the values $y(k)$, for $k = 0, 1, 2$, and 3 , in part (b) are correct.
- Verify the results in part (e) using MATLAB.

2-32. Write the state equations for the observer canonical form of a system, shown in Figure 2-10, which has the transfer function

$$G(z) = \frac{b_{n-1}z^{n-1} + \cdots + b_1z + b_0}{z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0}$$

2-33. Let $\Phi(k)$ be the state transition matrix for the equations

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k)$$

Show that $\Phi(k)$ satisfies the difference equation

$$\Phi(k+1) = \mathbf{A}\Phi(k)$$

2-34. Given the system described by the state equations

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = [0 \quad 0 \quad 1] \mathbf{x}(k)$$

- Calculate the transfer function $Y(z)/U(z)$, using (2-84).
- Draw a simulation diagram for this system, from the state equations given.
- Use Mason's gain formula and the simulation diagram to verify the transfer function found in part (a).
- Verify the results in part (a) using MATLAB.

2-35. Show that for the similarity transformation of (2-71),

$$\mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + \mathbf{D} = \mathbf{C}_w[z\mathbf{I} - \mathbf{A}_w]^{-1}\mathbf{B}_w + \mathbf{D}_w$$

2-36. Section 2.7 gives some standard forms for state equations (simulation diagrams). The MATLAB statement

$$[\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}] = \text{tf2ss}(\text{num}, \text{den})$$

generates a standard set of state equations for the transfer function whose numerator coefficients are given in the vector *num* and denominator coefficients in the vector *den*.

- (a) Use the MATLAB statement given to generate a set of state equations for the transfer function

$$G(z) = \frac{3z + 4}{z^2 + 5z + 6}$$

- (b) Draw a simulation diagram for the state equations in part (a).
(c) Determine if the simulation diagram in part (b) is one of the standard forms in Section 2.7.

Sampling and Reconstruction

3.1 INTRODUCTION

In Chapter 2 the concept of a discrete system was developed. We found that a discrete system is described (modeled) by a difference equation and that signals within the system are described by number sequences (e.g., $\{e(k)\}$). Some of these number sequences may be generated by sampling a continuous time signal (e.g., in digital control systems). To provide a basis for thoroughly understanding the operation of digital control systems, it is necessary to determine the effects of sampling a continuous-time signal. These topics are investigated in this chapter.

Sections 3.8 and 3.9 are devoted to the internal operation of digital-to-analog (D/A) and analog-to-digital (A/D) converters. A background in electrical engineering is needed to understand much of this material. However, the nonelectrical engineer will be able to understand the characteristics of different types of A/D and D/A converters by reading these sections.

3.2 SAMPLED-DATA CONTROL SYSTEMS

In this section the type of sampling that generally occurs in sampled-data control systems and in digital control systems is introduced, and a mathematical model of this sampling is developed. From this model we may determine the effects of the sampling on the information content of the signal that is sampled.

To introduce sampled-data systems, we consider the radar tracking system of Figure 3-1a. This system is described in Section 1.5. We consider only the control

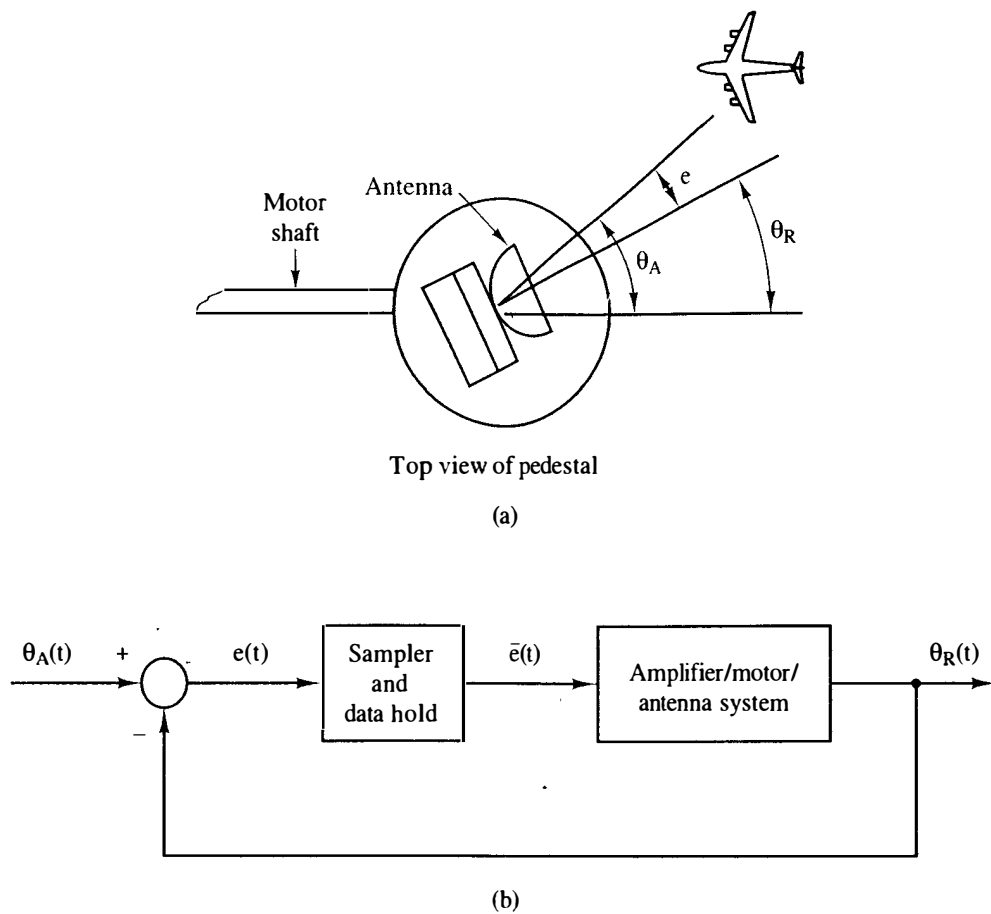


Figure 3-1 Sampled-data control system.

of the yaw angle $\theta_R(t)$, shown in the top view of the pedestal. The closed-loop system is to track the aircraft shown automatically. In Figure 3-1a, $\theta_R(t)$ is the yaw-axis pointing angle of the antenna, and $\theta_A(t)$ is the angle to the aircraft. Hence the tracking error is $e(t)$, given by

$$e(t) = \theta_A(t) - \theta_R(t)$$

Assume that the radar transmits every T seconds. Then the error $e(t)$ is known only every T seconds. The block diagram of this system is shown in Figure 3-1b. The radar receiver must output a voltage at every instant of time to the power amplifier. Since only $e(kT)$ is known, a decision must be made as to the form of power amplifier input, $\bar{e}(t)$, for $t \neq kT$.

In general, it is undesirable to apply a signal in sampled form, such as a train of narrow rectangular pulses, as shown in Figure 3-2, to a plant, because of the high-frequency components inherently present in that signal. Therefore, a data-reconstruction device, called a data hold, is inserted into the system directly following the sampler. The purpose of the data hold is to reconstruct the sampled signal into a form that closely resembles the signal before sampling. The simplest data-reconstruction device, and by far the most common one, is the zero-order hold. The

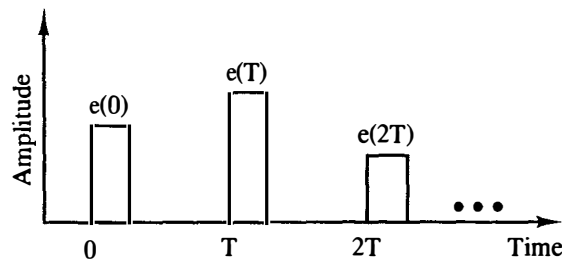


Figure 3-2 Sampled signal in pulse form.

operation of a sampler/zero-order hold combination is described by the signals shown in Figure 3-3. The zero-order hold clamps the output signal to a value equal to that of the input signal at the sampling instant.

The sampler and zero-order hold can be represented in block diagram form as shown in Figure 3-4. The signal $\bar{e}(t)$ can be expressed as

$$\begin{aligned} \bar{e}(t) = & e(0)[u(t) - u(t - T)] + e(T)[u(t - T) - u(t - 2T)] \\ & + e(2T)[u(t - 2T) - u(t - 3T)] + \dots \end{aligned} \quad (3-1)$$

where $u(t)$ is the unit-step function. The Laplace transform of $\bar{e}(t)$ is $\bar{E}(s)$, given by

$$\begin{aligned} \bar{E}(s) = & e(0)\left[\frac{1}{s} - \frac{\epsilon^{-Ts}}{s}\right] + e(T)\left[\frac{\epsilon^{-Ts}}{s} - \frac{\epsilon^{-2Ts}}{s}\right] \\ & + e(2T)\left[\frac{\epsilon^{-2Ts}}{s} - \frac{\epsilon^{-3Ts}}{s}\right] + \dots \\ = & \left[\frac{1 - \epsilon^{-Ts}}{s}\right][e(0) + e(T)\epsilon^{-Ts} + e(2T)\epsilon^{-2Ts} + \dots] \\ = & \left[\sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs}\right]\left[\frac{1 - \epsilon^{-Ts}}{s}\right] \end{aligned} \quad (3-2)$$

(The Laplace transform is reviewed in Appendix VII, and Appendix VIII gives a table of Laplace transforms.)

The first factor in the last expression in (3-2) is seen to be a function of the input signal $e(t)$ and the sampling period T . The second factor is seen to be independent of $e(t)$ and therefore can be considered to be a transfer function. Thus the sampler/

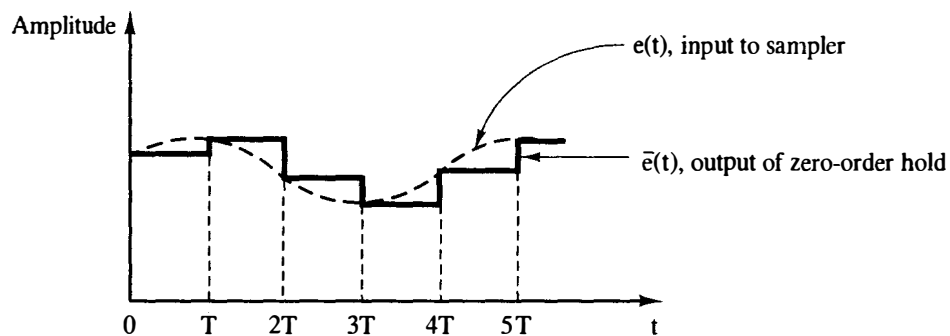


Figure 3-3 Input and output signals of sampler/data hold.

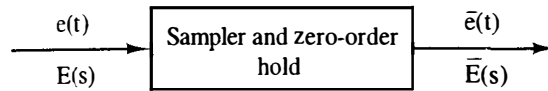


Figure 3-4 Sampler and data hold.

hold operation can be represented as shown in Figure 3-4. The function $E^*(s)$, called the starred transform, is *defined* as

$$E^*(s) = \sum_{n=0}^{\infty} e(nT) e^{-nTs} \quad (3-3)$$

Hence (3-2) is satisfied by the representation in Figure 3-5. The operation denoted by the switch in Figure 3-5 is defined by (3-3) and is called an ideal sampler; the operation denoted by the transfer function is called the data hold. It is to be emphasized that $E^*(s)$ *does not appear in the physical system* but appears as a result of factoring (3-2). The sampler (switch) in Figure 3-5 does not model a physical sampler and the block does not model a physical data hold. However, the combination does accurately model the input-output characteristics of the sampler-data hold device, as demonstrated earlier.

The operation symbolized by the sampler in Figure 3-5 cannot be represented by a transfer function. From (3-3) we see that the output of the sampler is a function of $e(t)$ only at $t = kT$, $k = 0, 1, 2, \dots$. Hence many different input signals can result in the same output signal $E^*(s)$. However, the representation of a sampler as a transfer function would require each different $E(s)$ to result in a different $E^*(s)$. Hence no transfer function exists for the sampler, and this property of the sampler complicates the analysis of systems of the type shown in Figure 3-1b.

As an aside, many of the systems that we consider will have unity gain in the feedback path, as shown in Figure 3-1b. In practical systems, the sensor transfer function will appear in the feedback path. However, the bandwidth of the sensor is usually much greater than that of the plant, and thus the sensor can be considered to be a constant gain for system analysis and design. For a discussion of the problems in converting practical control systems to systems with unity gain in the feedback path, see Ref. 1.

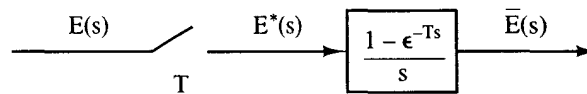


Figure 3-5 Representation of sampler and data hold.

3.3 THE IDEAL SAMPLER

The inverse Laplace transform of $E^*(s)$ is, from (3-3),

$$\begin{aligned} e^*(t) = \mathcal{L}^{-1}[E^*(s)] &= e(0)\delta(t) + e(T)\delta(t - T) \\ &+ e(2T)\delta(t - 2T) + \dots \end{aligned} \quad (3-4)$$

where $\delta(t)$ is the unit impulse function occurring at $t = 0$. Then $e^*(t)$ is a train of

impulse functions whose weights are equal to the values of the signal at the instants of sampling. Thus $e^*(t)$ can be represented as shown in Figure 3-6, since the impulse function has infinite amplitude at the instant it occurs. Note again that $e^*(t)$ is not a physical signal.

The sampler that appears in a sampler/hold model is usually referred to as an *ideal sampler*, since nonphysical signals (impulse functions) appear on its output. This sampler is also referred to as an *impulse modulator*. To demonstrate this modulation concept, we define

$$\delta_T(t) = \sum_{n=0}^{\infty} \delta(t - nT) = \delta(t) + \delta(t - T) + \dots \quad (3-5)$$

Then $e^*(t)$ can be expressed as

$$\begin{aligned} e^*(t) &= e(t)\delta_T(t) = e(t)\delta(t) + e(t)\delta(t - T) + \dots \\ &= e(0)\delta(t) + e(T)\delta(t - T) + \dots \end{aligned} \quad (3-6)$$

In this form it can be readily seen that $\delta_T(t)$ is the carrier in the modulation process, and $e(t)$ is the modulating signal [2]. The Laplace transform requires $e(t)$ to be zero for $t < 0$ [3]. For this reason the summation in (3-5) can be taken from $n = -\infty$ to $n = \infty$ with no change in (3-6). Two equivalent representations of the ideal sampler are given in Figure 3-7.

A problem arises in the definition of the ideal sampler output in (3-4) if $e(t)$ has a discontinuity at $t = kT$. For example, if $e(t)$ is a unit-step function, what value is used for $e(0)$ in (3-4)? In order to be consistent in the consideration of discontinuous signals, the output signal of an ideal sampler is *defined* as follows:

Definition. The output signal of an ideal sampler is defined as the signal whose Laplace transform is

$$E^*(s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs} \quad (3-7)$$

where $e(t)$ is the input signal to the sampler. If $e(t)$ is discontinuous at $t = kT$, where k is an integer, then $e(kT)$ is taken to be $e(kT^+)$. The notation $e(kT^+)$ indicates the value of $e(t)$ as t approaches kT from the right (i.e., at $t = kT + \epsilon$, where ϵ is made arbitrarily small).

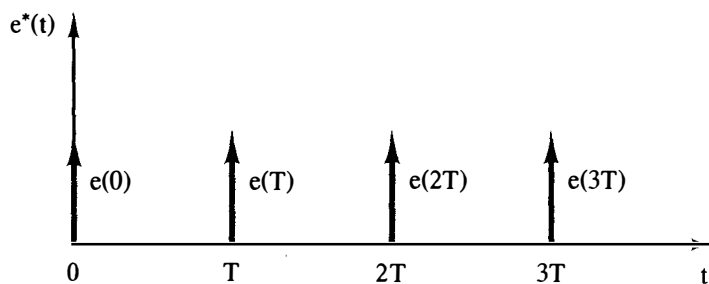


Figure 3-6 Representation of $e^*(t)$.

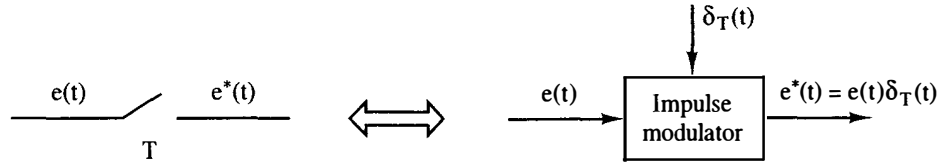


Figure 3-7 Representations of the ideal sampler.

It is important to note that the definition of the sampling operation as specified in (3-7) together with the zero-order-hold transfer function defined by

$$G_{ho}(s) = \frac{1 - e^{-Ts}}{s} \quad (3-8)$$

yield the correct mathematical description of the sampler/hold operation defined by (3-2). It should also be noted that if the signal to be sampled contains an impulse function at a sampling instant, the Laplace transform of the sampled signal does not exist; but since continuous signals in practical situations never contain impulse functions, this limitation is of no practical concern.

Example 3.1

Determine $E^*(s)$ for $e(t) = u(t)$, the unit step. For the unit step, $e(nT) = 1$, $n = 0, 1, 2, \dots$. Thus from (3-7),

$$E^*(s) = \sum_{n=0}^{\infty} e(nT)e^{-nTs} = e(0) + e(T)e^{-Ts} + e(2T)e^{-2Ts} + \dots$$

or

$$E^*(s) = 1 + e^{-Ts} + e^{-2Ts} + \dots$$

$E^*(s)$ can be expressed in closed form using the following relationship. For $|x| < 1$,

$$\frac{1}{1-x} = 1 + x + x^2 + \dots$$

The condition $|x| < 1$ guarantees convergence of the series. Hence the expression for $E^*(s)$ above can be written in closed form as

$$E^*(s) = \frac{1}{1 - e^{-Ts}}, \quad |e^{-Ts}| < 1$$

Example 3.2

Determine $E^*(s)$ for $e(t) = e^{-t}$. (From 3-7),

$$\begin{aligned} E^*(s) &= \sum_{n=0}^{\infty} e(nT)e^{-nTs} \\ &= 1 + e^{-T}e^{-Ts} + e^{-2T}e^{-2Ts} + \dots \\ &= 1 + e^{-(1+s)T} + (e^{-(1+s)T})^2 + \dots \\ &= \frac{1}{1 - e^{-(1+s)T}}, \quad |e^{-(1+s)T}| < 1 \end{aligned}$$

3.4 EVALUATION OF $E^*(s)$

$E^*(s)$, as defined by (3-7), has limited usefulness in analysis because it is expressed as an infinite series. However, for many useful time functions, $E^*(s)$ can be expressed in closed form. In addition, there is a third form of $E^*(s)$ that is useful. These two additional forms of $E^*(s)$ will now be investigated.

As was seen in Section 3.3, we can express the inverse Laplace transform of $E^*(s)$ as [see (3-6)]

$$e^*(t) = e(t)\delta_T(t) \quad (3-9)$$

If we then take the Laplace transform of $e^*(t)$ using the complex convolution integral [3], we can derive two additional expressions for $E^*(s)$. These derivations are given in Appendix III. The resultant expressions are

$$E^*(s) = \sum_{\substack{\text{at poles} \\ \text{of } E(\lambda)}} \left[\text{residues of } E(\lambda) \frac{1}{1 - e^{-T(s-\lambda)}} \right] \quad (3-10)$$

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) + \frac{e(0)}{2} \quad (3-11)$$

where ω_s is the radian sampling frequency, that is, $\omega_s = 2\pi/T$. The residue of a function is defined in Section 2.6. The expression (3-10) is useful in generating tables for the starred transform $E^*(s)$, and expression (3-11) will prove to be useful in analysis, as illustrated in the next section. Some examples that illustrate the use of (3-10) will now be given.

Example 3.3

Determine $E^*(s)$ given that

$$E(s) = \frac{1}{(s+1)(s+2)}$$

From (3-10),

$$E(\lambda) \frac{1}{1 - e^{-T(s-\lambda)}} = \frac{1}{(\lambda+1)(\lambda+2)(1 - e^{-T(s-\lambda)})}$$

Then

$$\begin{aligned} E^*(s) &= \sum_{\substack{\text{poles of} \\ E(\lambda)}} \left[\text{residues of } E(\lambda) \frac{1}{1 - e^{-T(s-\lambda)}} \right] \\ &= \frac{1}{(\lambda+2)(1 - e^{-T(s-\lambda)})} \Big|_{\lambda=-1} + \frac{1}{(\lambda+1)(1 - e^{-T(s-\lambda)})} \Big|_{\lambda=-2} \\ &= \frac{1}{1 - e^{-T(s+1)}} - \frac{1}{1 - e^{-T(s+2)}} \end{aligned}$$

Example 3.4

We wish to determine the starred transform of $e(t) = \sin \beta t$. The corresponding $E(s)$ is

$$E(s) = \frac{\beta}{s^2 + \beta^2} = \frac{\beta}{(s - j\beta)(s + j\beta)}$$

$E^*(s)$ can be evaluated from the expression

$$\begin{aligned} E^*(s) &= \sum_{\text{poles of } E(\lambda)} \left[\text{residues of } \frac{\beta}{(\lambda - j\beta)(\lambda + j\beta)(1 - e^{-T(s-\lambda)})} \right] \\ &= \frac{\beta}{(\lambda + j\beta)(1 - e^{-T(s-\lambda)})} \Big|_{\lambda = j\beta} + \frac{\beta}{(\lambda - j\beta)(1 - e^{-T(s-\lambda)})} \Big|_{\lambda = -j\beta} \\ &= \frac{1}{2j} \left[\frac{1}{1 - e^{-Ts} e^{j\beta T}} - \frac{1}{1 - e^{-Ts} e^{-j\beta T}} \right] \\ &= \frac{e^{-Ts} \sin \beta T}{1 - 2e^{-Ts} \cos \beta T + e^{-2Ts}} \end{aligned}$$

using the equations from Euler's relation:

$$\cos \beta T = \frac{e^{j\beta T} + e^{-j\beta T}}{2}; \quad \sin \beta T = \frac{e^{j\beta T} - e^{-j\beta T}}{2j}$$

Example 3.5

Given $e(t) = 1 - e^{-t}$, determine $E^*(s)$, first using (3-3) and then (3-10). Now,

$$E(s) = \frac{1}{s(s+1)}$$

From (3-3),

$$\begin{aligned} E^*(s) &= \sum_{n=0}^{\infty} e(nT) e^{-nTs} \\ &= \sum_{n=0}^{\infty} (1 - e^{-nT}) e^{-nTs} \\ &= \sum_{n=0}^{\infty} e^{-nTs} - \sum_{n=0}^{\infty} e^{-(1+s)nT} \\ &= \frac{1}{1 - e^{-Ts}} - \frac{1}{1 - e^{-(1+s)T}} \end{aligned}$$

From (3-10),

$$\begin{aligned} E^*(s) &= \sum_{\lambda=0, -1} \left[\text{residues of } \frac{1}{\lambda(\lambda+1)} \frac{1}{1 - e^{-T(s-\lambda)}} \right] \\ &= \frac{1}{1 - e^{-Ts}} - \frac{1}{1 - e^{-(1+s)T}} \end{aligned}$$

It is interesting to consider the case in which the function $e(t)$ contains a time delay. For example, consider a delayed signal of the type

$$e(t) = e_1(t - t_0)u(t - t_0)$$

From the shifting property of the Laplace transform (see Appendix VII),

$$E(s) = \epsilon^{-t_0 s} \mathcal{L}[e_1(t)] = \epsilon^{-t_0 s} E_1(s) \quad (3-12)$$

For this case, (3-10) does not apply; instead, special techniques are required to find the starred transform of a delayed signal in closed form. These techniques will be presented in Chapter 4, where the modified z-transform is developed. However, for the special case in which the time signal is delayed a whole number of sampling periods, (3-10) can be applied in a slightly different form,

$$[\epsilon^{-kTs} E_1(s)]^* = \epsilon^{-kTs} \sum_{\substack{\text{at poles} \\ \text{of } E_1(\lambda)}} \left[\text{residues of } E_1(\lambda) \frac{1}{1 - \epsilon^{-T(s-\lambda)}} \right] \quad (3-13)$$

where k is a positive integer (see Problem 3-7).

Example 3.6

The starred transform of $e(t) = [1 - \epsilon^{-(t-1)}]u(t-1)$, with $T = 0.5$ s, will now be found. First we find $E(s)$:

$$E(s) = \frac{\epsilon^{-s}}{s} - \frac{\epsilon^{-s}}{s+1} = \frac{\epsilon^{-s}}{s(s+1)}$$

Therefore, in (3-13), k is equal to 2 and

$$E_1(s) = \frac{1}{s(s+1)}$$

Then, from (3-13),

$$\begin{aligned} \left[\frac{\epsilon^{-s}}{s(s+1)} \right]^* &= \epsilon^{-s} \sum_{\lambda=0, -1} \left[\text{residues of } \frac{1}{\lambda(\lambda+1)(1 - \epsilon^{-0.5(s-\lambda)})} \right] \\ &= \epsilon^{-s} \left[\frac{1}{(\lambda+1)(1 - \epsilon^{-0.5(s-\lambda)})} \Big|_{\lambda=0} + \frac{1}{\lambda(1 - \epsilon^{-0.5(s-\lambda)})} \Big|_{\lambda=-1} \right] \\ &= \epsilon^{-s} \left[\frac{1}{1 - \epsilon^{-0.5s}} + \frac{-1}{1 - \epsilon^{-0.5(s+1)}} \right] = \frac{(1 - \epsilon^{-0.5})\epsilon^{-1.5s}}{(1 - \epsilon^{-0.5s})(1 - \epsilon^{-0.5(s+1)})} \end{aligned}$$

3.5 RESULTS FROM THE FOURIER TRANSFORM

In this section we present some results regarding the Fourier transform. These results are helpful in understanding the effects of sampling a signal.

The Fourier transform is defined by [4]

$$\mathcal{F}[e(t)] = E(j\omega) = \int_{-\infty}^{\infty} e(t)\epsilon^{-j\omega t} dt \quad (3-14)$$

In many texts the notation $E(\omega)$ is used instead of $E(j\omega)$; we discuss this further below.

Recall the unilateral Laplace transform from Appendix VII:

$$\mathcal{L}[e(t)] = E(s) = \int_0^{\infty} e(t) e^{-st} dt \quad (3-15)$$

If $e(t)$ is zero for $t < 0$, its Fourier transform is given by

$$\begin{aligned} \mathcal{F}[e(t)] &= E(j\omega) = \int_{-\infty}^0 e(t) e^{-j\omega t} dt + \int_0^{\infty} e(t) e^{-j\omega t} dt \\ &= \int_0^{\infty} e(t) e^{-j\omega t} dt = \mathcal{L}[e(t)] \Big|_{s=j\omega} \end{aligned}$$

provided that both transforms exist. This result can be expressed in general as

$$\mathcal{F}[e(t)u(t)] = \mathcal{L}[e(t)u(t)] \Big|_{s=j\omega} \quad (3-16)$$

Hence, for the case that $e(t)$ is zero for negative time, the Fourier transform of $e(t)$ is equal to the Laplace transform of $e(t)$ with s replaced with $j\omega$. We see then the reason for expressing the Fourier transform as $E(j\omega)$. In (3-16) it is assumed that both the Fourier transform and the Laplace transform exist. Recall that neither transform may exist for a given time function, or that the Laplace transform may exist while the Fourier transform does not.

A plot of the Fourier transform $E(j\omega)$ is called the *frequency spectrum* of $e(t)$. A common procedure for showing the frequency spectrum is to express $E(j\omega)$ as

$$E(j\omega) = |E(j\omega)| e^{j\theta(j\omega)} = |E(j\omega)| \angle \theta(j\omega)$$

and plot $|E(j\omega)|$ versus ω (the amplitude spectrum) and $\theta(j\omega)$ versus ω (the phase spectrum). We interpret the amplitude spectrum as giving the relative contents of a signal in different frequency bands. The meaning of the frequency spectrum of a signal is fundamental to understanding the transmission of signals through systems; those readers who do not have a good background in this area should review this material in a signals and systems book such as Ref. 4.

Consider now an analog system with the input signal $e(t)$, the output signal $y(t)$, and the transfer function $G(s)$; that is,

$$Y(s) = G(s)E(s) \quad (3-17)$$

If $e(t)$ is the unit impulse function $\delta(t)$, $E(s) = 1$ and $Y(s) = G(s)$. Hence $g(t) = \mathcal{L}^{-1}[G(s)]$ is the unit impulse response of the system with the transfer function $G(s)$. A physical system cannot respond to an input signal before this signal is applied. For this case, $g(t) = 0$ for $t < 0$. A system that satisfies this condition is called a *causal system*. Consequently, for a causal system, the Fourier transform of the unit impulse response is equal to the Laplace transform of that response with s replaced with $j\omega$, provided that the Fourier transform exists. Hence, for the system of (3-17), the Fourier transform yields

$$Y(j\omega) = G(j\omega)E(j\omega)$$

We see that $G(j\omega)$ determines the manner that the frequency spectrum of the input $e(t)$ is modified to yield the frequency spectrum of the output $y(t)$. The function $G(j\omega)$ is called the *frequency response*.

3.6 PROPERTIES OF $E^*(s)$

Two s -plane properties of $E^*(s)$ are now given. These properties are very important and will be used extensively in later derivations.

Property 1. $E^*(s)$ is periodic in s with period $j\omega_s$.

This property can be proved using either (3-3), (3-10), or (3-11). From (3-3),

$$E^*(s + jm\omega_s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nT(s + jm\omega_s)} \quad (3-18)$$

Since $\omega_s T = (2\pi/T)T = 2\pi$, and from Euler's relationship,

$$\epsilon^{j\theta} = \cos \theta + j \sin \theta$$

then

$$\epsilon^{-jnm\omega_s T} = \epsilon^{-jnm2\pi} = 1, \quad m \text{ an integer} \quad (3-19)$$

Thus, in (3-18),

$$E^*(s + jm\omega_s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs} = E^*(s) \quad (3-20)$$

Property 2. If $E(s)$ has a pole at $s = s_1$, then $E^*(s)$ must have poles at $s = s_1 + jm\omega_s$, $m = 0, \pm 1, \pm 2, \dots$

This property can be proved from (3-11). Consider $e(t)$ to be continuous at all sampling instants. Then

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) = \frac{1}{T} [E(s) + E(s + j\omega_s) + E(s + 2j\omega_s) + \dots + E(s - j\omega_s) + E(s - j2\omega_s) + \dots] \quad (3-21)$$

If $E(s)$ has a pole at $s = s_1$, then each term of (3-21) will contribute a pole at $s = s_1 + jm\omega_s$, where m is an integer.

It is important to note that no equivalent statement can be made concerning the zeros of $E^*(s)$; that is, the zero locations of $E(s)$ do not uniquely determine the zero locations of $E^*(s)$. However, the zero locations are periodic with period $j\omega_s$, as indicated by the first property of $E^*(s)$.

An example of pole-zero locations of $E^*(s)$ is given in Figure 3-8. The primary strip in the s -plane is defined as the strip for which $-\omega_s/2 \leq \omega \leq \omega_s/2$, as shown in Figure 3-8. Note that if the pole-zero locations are known for $E^*(s)$ in the primary strip, then the pole-zero locations in the entire s -plane are also known.

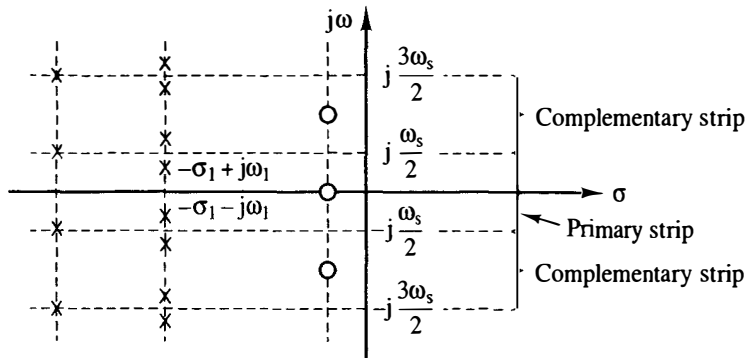


Figure 3-8 Pole-zero locations for $E^*(s)$.

In Figure 3-8, if $E(s)$ has a pole at $-\sigma_1 + j\omega_1$, the sampling operation will generate a pole in $E^*(s)$ at $-\sigma_1 + j(\omega_1 + \omega_s)$. Conversely, if $E(s)$ has a pole at $-\sigma_1 + j(\omega_1 + \omega_s)$, then $E^*(s)$ will have a pole at $-\sigma_1 + j\omega_1$. In fact, a pole location in $E(s)$ at $-\sigma_1 + j(\omega_1 + k\omega_s)$, k an integer, will result in identical pole locations in $E^*(s)$, regardless of the integer value of k .

The conclusion above can also be seen from the example of Figure 3-9. Note that both signals have the same starred transform, since the two signals have the same value at each sampling instant. Note also that $\omega_1 = \omega_s/4$, or $\omega_s = 4\omega_1$. Since

$$E_1(s) = \mathcal{L}[\cos \omega_1 t] = \frac{s}{s^2 + \omega_1^2} = \frac{s}{(s + j\omega_1)(s - j\omega_1)}$$

then $E_1(s)$ has a pole at $s = j\omega_1$. And since

$$E_2(s) = \mathcal{L}[\cos 3\omega_1 t] = \frac{s}{(s + j3\omega_1)(s - j3\omega_1)}$$

then $E_2(s)$ has a pole at $s = -j3\omega_1 = j(\omega_1 - \omega_s)$. The other pole of $E_1(s)$ occurs at $s = -j\omega_1$, and the other pole of $E_2(s)$ occurs at $s = j3\omega_1 = j(-\omega_1 + \omega_s)$.

The following discussion is based on the results of the Fourier transform given in the preceding section. Suppose that the signal $e(t)$ has the amplitude spectrum $|E(j\omega)|$ shown in Figure 3-10a, where $E(j\omega)$ is the Fourier transform of $e(t)$. Then $E^*(j\omega)$ has the amplitude spectrum shown in Figure 3-10b. This can be seen by evaluating (3-21) for $s = j\omega$.

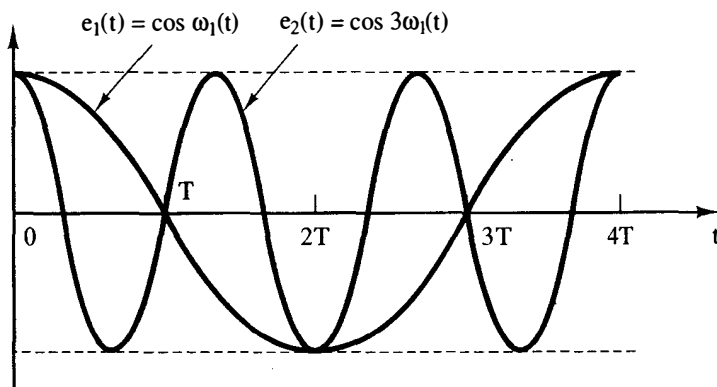


Figure 3-9 Two signals that have the same starred transform.

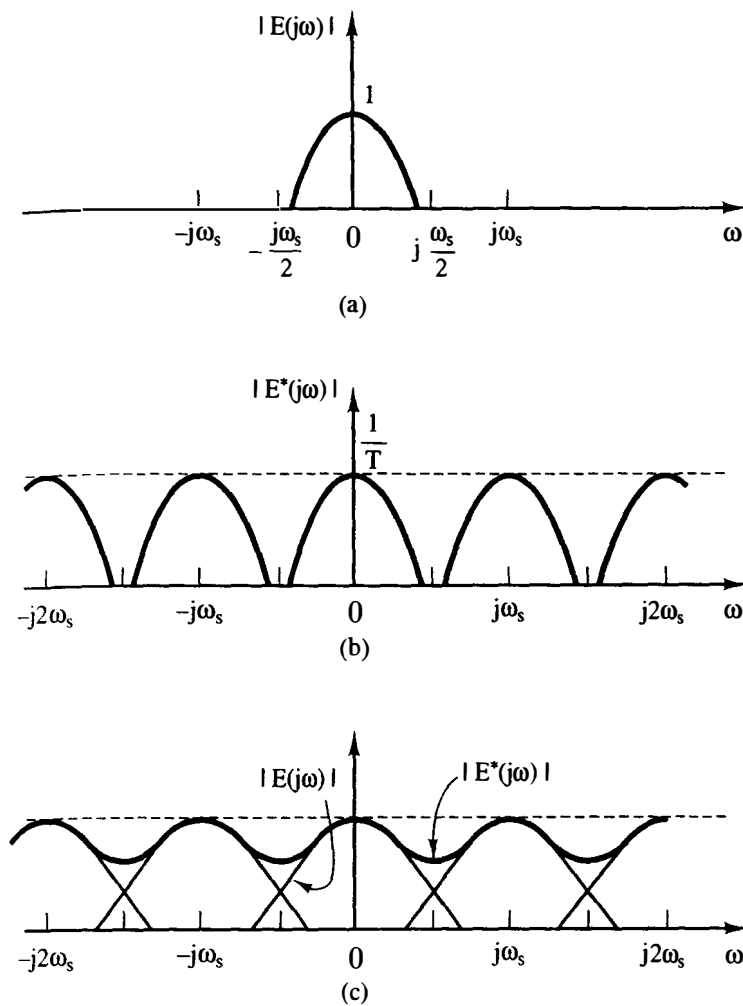


Figure 3-10 Frequency spectra for $E(j\omega)$ and $E^*(j\omega)$.

$$\begin{aligned}
 E^*(j\omega) = \frac{1}{T} [& E(j\omega) + E(j\omega + j\omega_s) + E(j\omega + 2j\omega_s) + \dots \\
 & + E(j\omega - j\omega_s) + E(j\omega - j2\omega_s) + \dots] + e(0)/2
 \end{aligned}
 \quad (3-22)$$

Hence the effect of ideal sampling is to replicate the original spectrum centered at ω_s , at $2\omega_s$, at $-\omega_s$, at $-2\omega_s$, and so on.

An ideal filter is a filter with a unity gain in the passband and zero gain outside the passband. Of course, such a filter is not physically realizable [7]. It is seen from Figure 3-10b that an ideal low-pass filter could completely recover $E(j\omega)[e(t)]$ if the bandwidth of the filter were $\omega_s/2$, for the case that the highest frequency present in $E(j\omega)$ is less than $\omega_s/2$. This is, of course, essentially a statement of Shannon's sampling theorem [5].

Shannon's Sampling Theorem. A function of time $e(t)$ which contains no frequency components greater than f_0 hertz is uniquely determined by the values of $e(t)$ at any set of sampling points spaced $1/(2f_0)$ seconds apart.

Suppose, in Figure 3-10b, that ω_s is decreased until the highest-frequency components present in $E(j\omega)$ are greater than $\omega_s/2$. Then $E^*(j\omega)$ has the amplitude spectrum shown in Figure 3-10c; and for this case, no filtering scheme, ideal or realizable, will recover $e(t)$. Thus, in choosing the sampling rate for a control system, the sampling frequency should be greater than twice the highest-frequency component of *significant amplitude* of the signal being sampled.

It is to be recalled that the ideal sampler is not a physical device, and thus the frequency spectrum, as shown in Figure 3-10, is not the spectrum of a signal that appears in a physical system. The ideas above will be extended to signals that do appear in physical systems after an investigation of data holds.

3.7 DATA RECONSTRUCTION

In most feedback control systems employing sampled data, a continuous signal is reconstructed from the sampled signal. The block diagram of a simple sampled-data control system is repeated in Figure 3-11. Suppose that the sampled signal is band-limited in frequency, such that the highest-frequency component of $e(t)$ is less than $\omega_s/2$. Then $E^*(j\omega)$ would have the frequency spectrum shown in Figure 3-10b, and theoretically the signal could be reconstructed exactly by employing an ideal low-pass filter. However, since ideal filters do not exist in physically realizable systems, we must employ approximations. Practical data holds are devices that approximate, in some sense, an ideal low-pass filter.

The reader may ask: Why discuss a sampled-data system as illustrated in Figure 3-11? The system contains a sampler to examine a continuous signal at discrete instants of time. Then a data hold is employed to try to reconstruct the original signal!

Our reply is that many existing control systems actually operate in this manner due to hardware implementation techniques. More important, we will later add a digital compensator block between the sampler and data-hold device in order to improve system performance, and the sampler is required in this case. In either case, our discussion of data-hold devices in this chapter is prerequisite to closed-loop system analysis and synthesis.

A commonly used method of data reconstruction is polynomial extrapolation. Using a Taylor's series expansion about $t = nT$, we can express $e(t)$ as

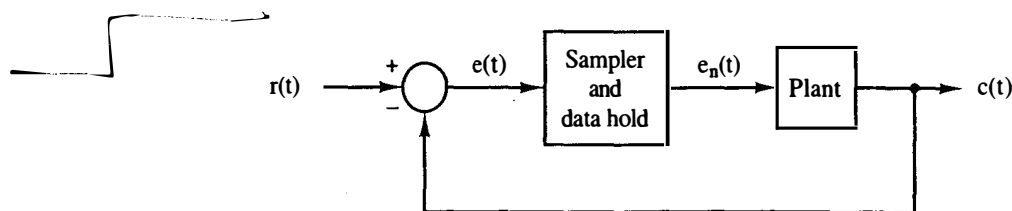


Figure 3-11 Sampled-data control system.

$$e(t) = e(nT) + e'(nT)(t - nT) + \frac{e''(nT)}{2!}(t - nT)^2 + \dots \quad (3-23)$$

where the prime denotes the derivative. $e_n(t)$ is defined as the reconstructed version of $e(t)$ for the n th sample period; that is,

$$e_n(t) \cong e(t) \quad \text{for } nT \leq t < (n + 1)T \quad (3-24)$$

Thus $e_n(t)$ denotes the output of the data hold. Since $e(t)$ enters the data hold only in sampled form, the values of the derivatives are not known. However, the derivatives may be approximated by the backward difference

$$e'(nT) = \frac{1}{T}[e(nT) - e[(n - 1)T]] \quad (3-25)$$

$$e''(nT) = \frac{1}{T}[e'(nT) - e'[(n - 1)T]] \quad (3-26)$$

and so on. Note that by substituting (3-25) into (3-26), we obtain

$$e''(nT) = \frac{1}{T} \left[\frac{1}{T}[e(nT) - e[(n - 1)T]] - \frac{1}{T}[e[(n - 1)T] - e[(n - 2)T]] \right]$$

or

$$e''(nT) = \frac{1}{T^2}[e(nT) - 2e[(n - 1)T] + e[(n - 2)T]] \quad (3-27)$$

Three types of data hold based on the foregoing relationships—the zero-, first-, and fractional-order holds—will now be discussed.

Zero-Order Hold

If only the first term in the expansion of (3-23) is used, the data hold is called a zero-order hold. Here we assume that the function $e(t)$ is approximately constant within the sampling interval at a value equal to that of the function at the preceding sampling instant. Therefore, for the zero-order hold,

$$e_n(t) = e(nT), \quad nT \leq t < (n + 1)T \quad (3-28)$$

Recall that the $e_n(t)$ for a zero-order hold has been defined as $\bar{e}(t)$ in (3-1). Note that no memory is required, and therefore this data hold is the simplest to construct. The transfer function of the zero-order hold was derived in (3-2). However, this transfer function will be derived again using a simpler technique—one that may be easily used in the derivations of transfer functions for other data holds. Using the model of Figure 3-5 for the sampler-data-hold device, we note that the input to the data hold will be only impulse functions. The signal $e_o(t)$ shown in Figure 3-12 then describes the data hold output if the input $e_i(t)$ to the data hold is a unit impulse function. Thus

$$e_o(t) = u(t) - u(t - T)$$

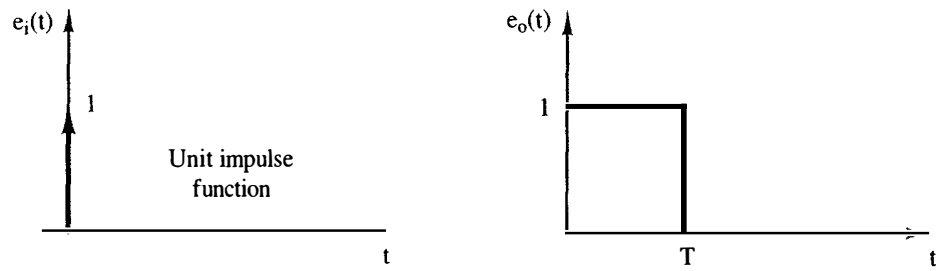


Figure 3-12 Input and output signals for the zero-order hold.

and

$$E_o(s) = \frac{1}{s} - \frac{e^{-Ts}}{s}$$

Since $E_i(s) = 1$, the transfer function of the zero-order hold is

$$G_{h0}(s) = \frac{E_o(s)}{E_i(s)} = \frac{1 - e^{-Ts}}{s} \quad (3-29)$$

as has already been shown in (3-2). In addition, recall that (3-29) is not the transfer function of a physical device since this equation was derived assuming that impulse functions occur at the input to the zero-order hold. Thus, as shown earlier in this chapter, if (3-7) is used to describe the sampling operation mathematically, and if (3-29) is used in conjunction with (3-7) to describe the data hold mathematically, a correct mathematical model of the overall sample-hold operation is obtained.

To obtain the frequency response of the zero-order hold, consider the following development:

$$\begin{aligned} G_{h0}(j\omega) &= \frac{1 - e^{-j\omega T}}{j\omega} e^{j(\omega T/2)} e^{-j(\omega T/2)} = \frac{2e^{-j(\omega T/2)}}{\omega} \left[\frac{e^{j(\omega T/2)} - e^{-j(\omega T/2)}}{2j} \right] \\ &= T \frac{\sin(\omega T/2)}{\omega T/2} e^{-j(\omega T/2)} \end{aligned} \quad (3-30)$$

Since

$$\frac{\omega T}{2} = \frac{\omega}{2} \left(\frac{2\pi}{\omega_s} \right) = \frac{\pi\omega}{\omega_s}$$

(3-30) can be expressed as

$$G_{h0}(j\omega) = T \frac{\sin(\pi\omega/\omega_s)}{\pi\omega/\omega_s} e^{-j(\pi\omega/\omega_s)} \quad (3-31)$$

Thus

$$|G_{h0}(j\omega)| = T \left| \frac{\sin(\pi\omega/\omega_s)}{\pi\omega/\omega_s} \right| \quad (3-32)$$

and

$$\angle G_{h0}(j\omega) = -\frac{\pi\omega}{\omega_s} + \theta, \quad \theta = \begin{cases} 0, & \sin\left(\frac{\pi\omega}{\omega_s}\right) > 0 \\ \pi, & \sin\left(\frac{\pi\omega}{\omega_s}\right) < 0 \end{cases} \quad (3-33)$$

The amplitude and phase plots for $G_{h0}(j\omega)$ are shown in Figure 3-13.

A word is in order concerning the interpretation of the frequency response of the zero-order hold. First, it must be remembered that the data hold must be preceded by an ideal sampler. Now, suppose that a sinusoid of frequency ω_1 is applied to the ideal sampler, where $\omega_1 < \omega_s/2$. Recall that the Fourier transform of $e(t) = 2 \cos \omega_1 t$ is given by

$$E(j\omega) = \mathcal{F}[2 \cos \omega_1 t] = \delta(\omega - \omega_1) + \delta(\omega + \omega_1)$$

The frequency spectrum of this sinusoid is then two unit impulse functions, as shown in Figure 3-14a. In this figure, the lengths of the arrows representing the impulse functions denotes the weights. Then, from (3-22), the output of the sampler contains the frequencies in the time domain represented by the impulse functions in the

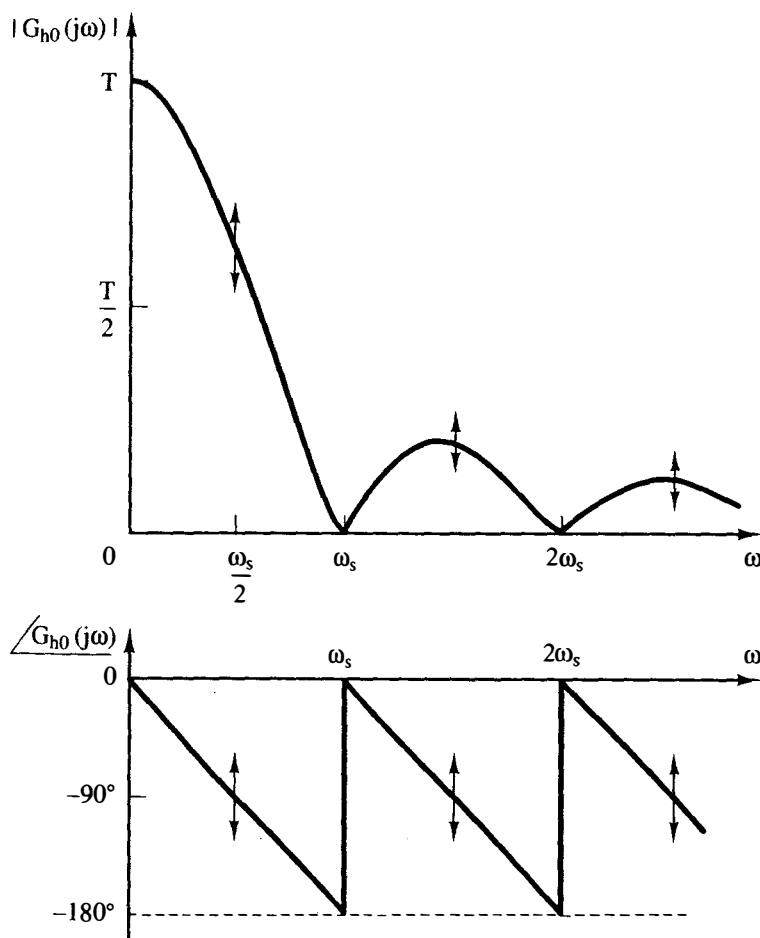


Figure 3-13 Frequency response of the zero-order hold.

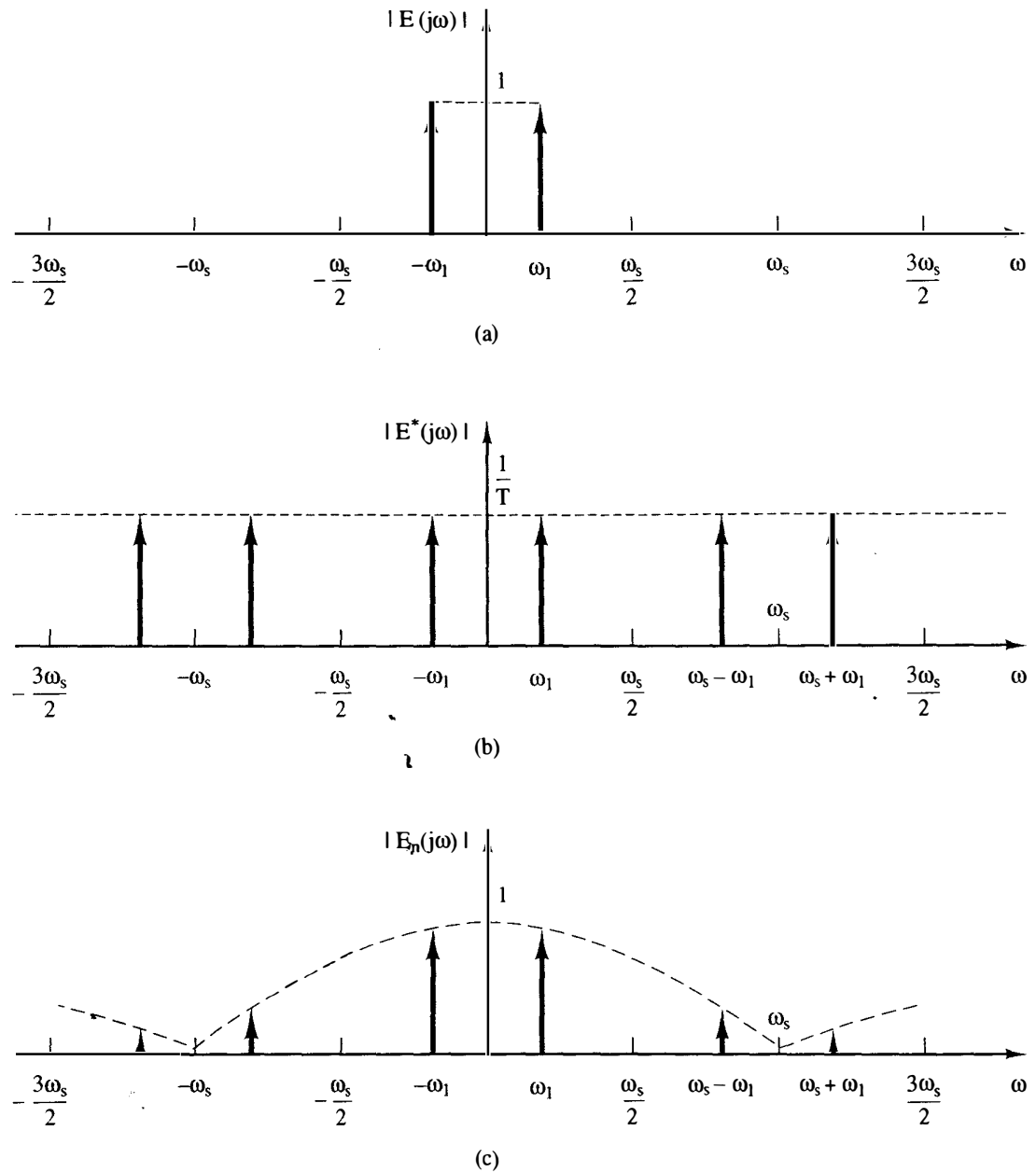


Figure 3-14 Sinusoidal response of the sampler and zero-order hold; (a) input signal to the ideal sampler; (b) output signal of the ideal sampler; (c) output signal from the zero-order hold.

frequency domain, as shown in Figure 3-14b. Thus the frequency response of the zero-order hold may be used to determine the amplitude spectrum of the data-hold output signal. The output signal components are shown in Figure 3-14c. Hence a signal of the type shown in Figure 3-15 has the frequency spectrum of Figure 3-14c.

Note that the output signal amplitude spectrum will be the same as that shown in Figure 3-14c if the input signal frequency is $\omega = k\omega_s \pm \omega_1$, $k = 0, 1, 2, 3, \dots$

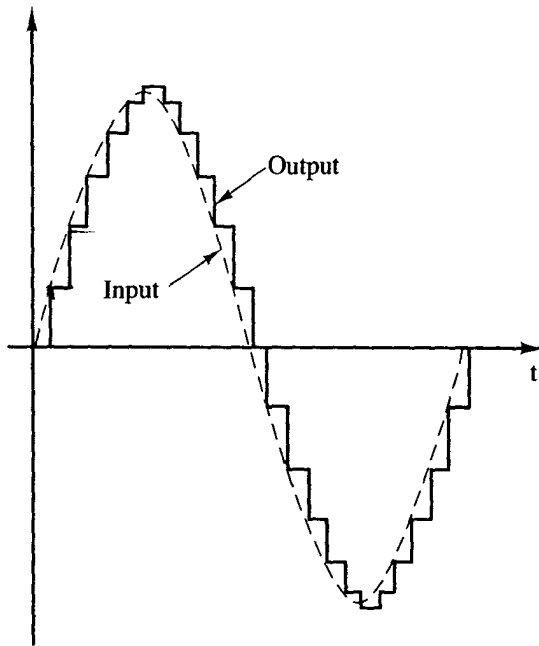


Figure 3-15 Output of a sampler/zero-order hold for a sinusoidal input.

Hence any frequencies $\omega > \omega_s/2$ will reflect into the frequency range $0 < \omega < \omega_s/2$. This effect is called *frequency foldover*, or *frequency aliasing*. These reflected frequencies will be interpreted by the system as low-frequency information in $e(t)$, which generally cannot be tolerated. The frequency aliasing can be prevented either by increasing ω_s or by placing an analog *antialiasing* filter in front of the sampler. The antialiasing filter is a low-pass filter that removes any frequencies in $e(t)$ that are greater than $\omega_s/2$. Since low-pass filters introduce phase lag, the cutoff frequency of the antialiasing filter cannot be made so low as to destabilize the control system. As a final point, note that, in Figure 3-14, when $\omega_1 \ll \omega_s/2$, the high-frequency components of $E^*(j\omega)$ will occur near zeros of $G_{h0}(j\omega)$; hence the sampler and zero-order hold will have little effect on the signal.

Suppose that the frequency ω_1 , in Figure 3-14, is equal to $\omega_s/2$. In this case the frequency components ω_1 and $\omega_s - \omega_1$ are superimposed, and the data-hold output is a function of the phase of the input sinusoid. Note that the amplitude of the data-hold output can range from 0 to a value greater than the amplitude of the input signal. We emphasize this effect by showing arrows at $k\omega_s + \omega_s/2$, $k = 0, 1, 2, \dots$, in Figure 3-13.

First-Order Hold

The first two terms of (3-23) are used to realize the first-order hold. Therefore,

$$e_n(t) = e(nT) + e'(nT)(t - nT), \quad nT \leq t < (n+1)T \quad (3-34)$$

where, from (3-25),

$$e'(nT) = \frac{e(nT) - e[(n-1)T]}{T} \quad (3-35)$$

This expression indicates that the extrapolated function within a given interval is a straight line and that its slope is determined by the values of the function at the sampling instants in the previous interval. Note that memory is required in the realization of this data hold, since $e[(n-1)T]$ must be available at $t = nT$.

To determine the transfer function of a first-order hold, assume that the input is a unit impulse function. Then, from (3-34) and (3-35), the output, shown in Figure 3-16a, is

$$e_o(t) = u(t) + \frac{1}{T}tu(t) - 2u(t-T) - \frac{2}{T}(t-T)u(t-T) + u(t-2T) + \frac{1}{T}(t-2T)u(t-2T)$$

Since $E_i(s) = 1$,

$$G_{hl}(s) = \frac{E_o(s)}{E_i(s)} = \frac{1}{s} - \frac{2\epsilon^{-Ts}}{s} + \frac{\epsilon^{-2Ts}}{s} + \frac{1}{Ts^2}(1 - 2\epsilon^{-Ts} + \epsilon^{-2Ts})$$

or

$$G_{hl}(s) = \frac{1+Ts}{T} \left[\frac{1-\epsilon^{-Ts}}{s} \right]^2 \quad (3-36)$$

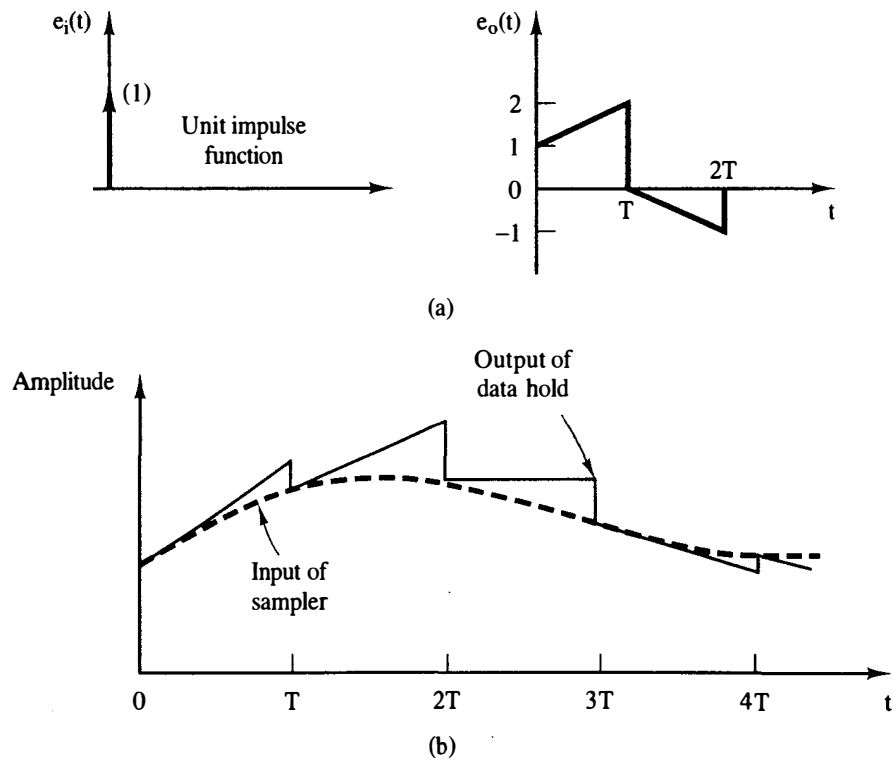


Figure 3-16 Example of the output of a first-order hold.

An example of the input-output waveforms of a sampler and first-order hold is shown in Figure 3-16b.

The frequency response of the first-order hold is obtained from (3-36).

$$G_{h1}(j\omega) = \frac{1 + j\omega T}{T} \left[\frac{1 - e^{-j\omega T}}{j\omega} \right]^2$$

$$|G_{h1}(j\omega)| = T \sqrt{1 + \frac{4\pi^2 \omega^2}{\omega_s^2} \left[\frac{\sin(\pi\omega/\omega_s)}{\pi\omega/\omega_s} \right]^2} \quad (3-37)$$

$$\angle G_{h1}(j\omega) = \tan^{-1} \left(\frac{2\pi\omega}{\omega_s} \right) - \frac{2\pi\omega}{\omega_s} \quad (3-38)$$

The amplitude and phase characteristics of the first-order hold are shown in Figure 3-17. Note that the first-order hold provides a better approximation of the ideal low-pass filter in the vicinity of zero frequency than does the zero-order hold. However, for larger ω , the zero-order hold yields a better approximation. Consider once again the sideband frequencies generated by the sampling process, as illustrated in Figure 3-14b. If $\omega_1 \ll \omega_s/2$, the first-order hold provides better reconstruction of the sampled signal than does the zero-order hold. However, if ω_1 is of the same order of magnitude as $\omega_s/2$, the zero-order hold may yield better results in the reconstruction process. Thus in some applications the zero-order hold is superior to the

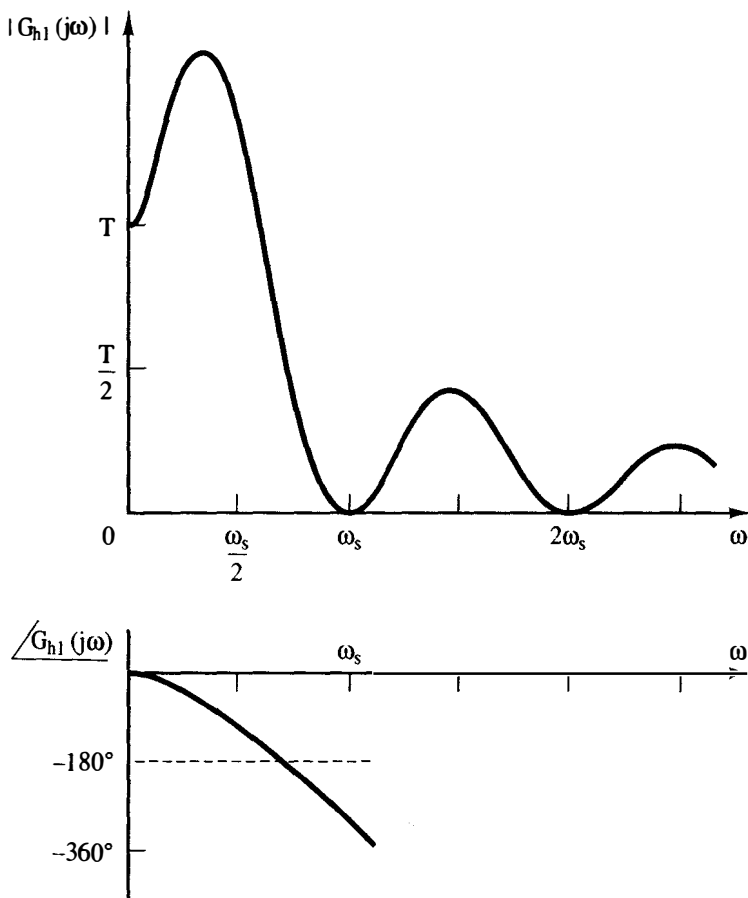


Figure 3-17 Frequency response of the first-order hold.

first-order hold. However, in any case, the zero-order hold is by far the most commonly used device, because of cost considerations.

Fractional-Order Holds

When using the first-order hold we essentially perform a linear extrapolation from one sampling interval to the next; that is, we assume that by using the approximate slope of the signal in the interval from $(n - 1)T$ to nT we can obtain the value of the signal at $(n + 1)T$. The error generated in this process can be reduced by using only a fraction of the slope in the previous interval, as shown in Figure 3-18. In this figure it is assumed that the input is a unit impulse and the value of k ranges from zero to unity. The frequency response of this data hold is shown in Figure 3-19. Note that for $k = 0$, the hold is a zero-order hold; for $k = 1$, a first-order hold is obtained. Although it is difficult to determine the optimum value of k except in certain specific

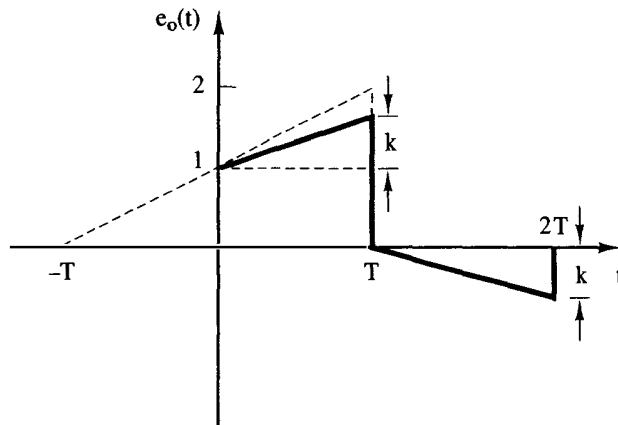


Figure 3-18 Impulse response of the fractional-order hold.

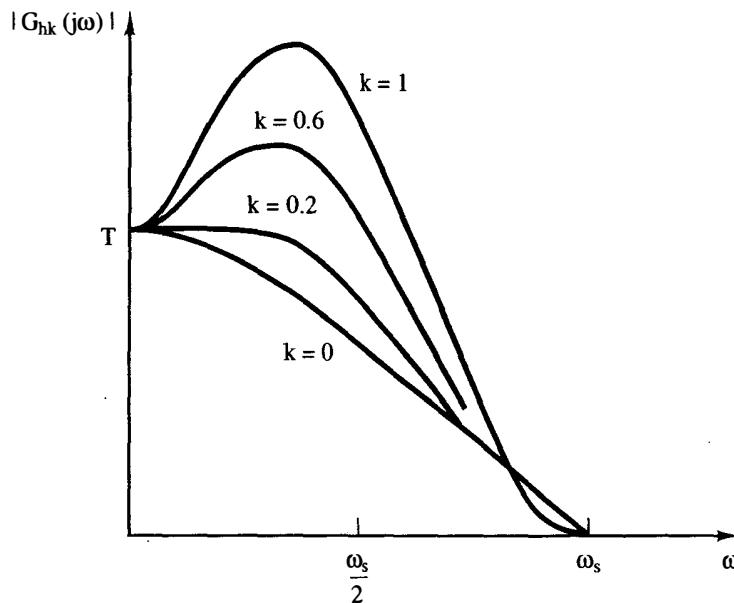


Figure 3-19 Frequency response magnitudes for the fractional-order hold.

circumstances, the fractional-order hold may be used to match the data-hold frequency response to the sampled signal's frequency spectrum, thereby generating minimum error extrapolations. By using the techniques given above, the transfer function of the fractional-order hold can be shown to be

$$G_{hk}(s) = (1 - k\epsilon^{-Ts}) \frac{1 - \epsilon^{-Ts}}{s} + \frac{k}{Ts^2} (1 - \epsilon^{-Ts})^2 \quad (3-39)$$

The derivation of this transfer function is given as an exercise in Problem 3-22.

3.8 DIGITAL-TO-ANALOG CONVERSION*

This section and Section 3.9 describe the most common methods for digital-to-analog and analog-to-digital conversion. Although the circuit diagrams have been simplified, the functional nature of each method is preserved. Upon completion of the sections the reader should be able to understand the literature on these subjects.

The basic function of the digital-to-analog converter (DAC or D/A) is to convert a digital representation of a number into its equivalent analog voltage. The output voltage of the D/A converter can be represented as

$$V_o = V_{fs}[A_1 2^{-1} + A_2 2^{-2} + \cdots + A_n 2^{-n}] \quad (3-40)$$

V_{fs} represents a reference voltage which determines the full-scale output voltage of the converter, and A_1 through A_n represent the binary digits or bits of the input word. A_1 is called the most significant bit (MSB) and corresponds to a voltage of $V_{fs}/2$. A_n is the least significant bit (LSB) and corresponds to $V_{fs}/2^n$. For the sake of our discussion, an "on" bit equals 1 and an "off" bit equals 0. The *resolution* of the converter is the smallest analog change that can be produced by the converter and is equal to the value of the LSB in volts. However, it is also often specified as a percentage of full scale or just as n -bit resolution.

One of the simplest DAC circuits is given in Figure 3-20a and uses the summing amplifier [6] and weighted resistor network. The input binary word controls the switches with an on bit indicating a closed switch and an off bit indicating an open switch. The resistors are weighted progressively by a factor of 2, thereby producing the desired binary weighted contributions to the output. Two problems arise. The first problem is that the DAC requires accurate resistor ratios to be maintained over a very wide range of resistor values (a range of $1024 - 1$ for a 10-bit DAC). Also, since the switches are in series with the resistors, this "on" resistance must be very low, and they should have zero offset voltage. These last two requirements can be met using good MOSFETs or JFETs as switches. However, the wide range of resistor values is not suitable for monolithic converters of moderate to high resolution.

The R - $2R$ ladder shown in Figure 3-20b avoids the problem of a wide range

* Section 3.8, contributed by Richard C. Jaeger, is based on his "Data Acquisition Systems," Tutorial Notes, IECI 81, San Francisco, Nov. 9, 1981, and is included here with his permission.

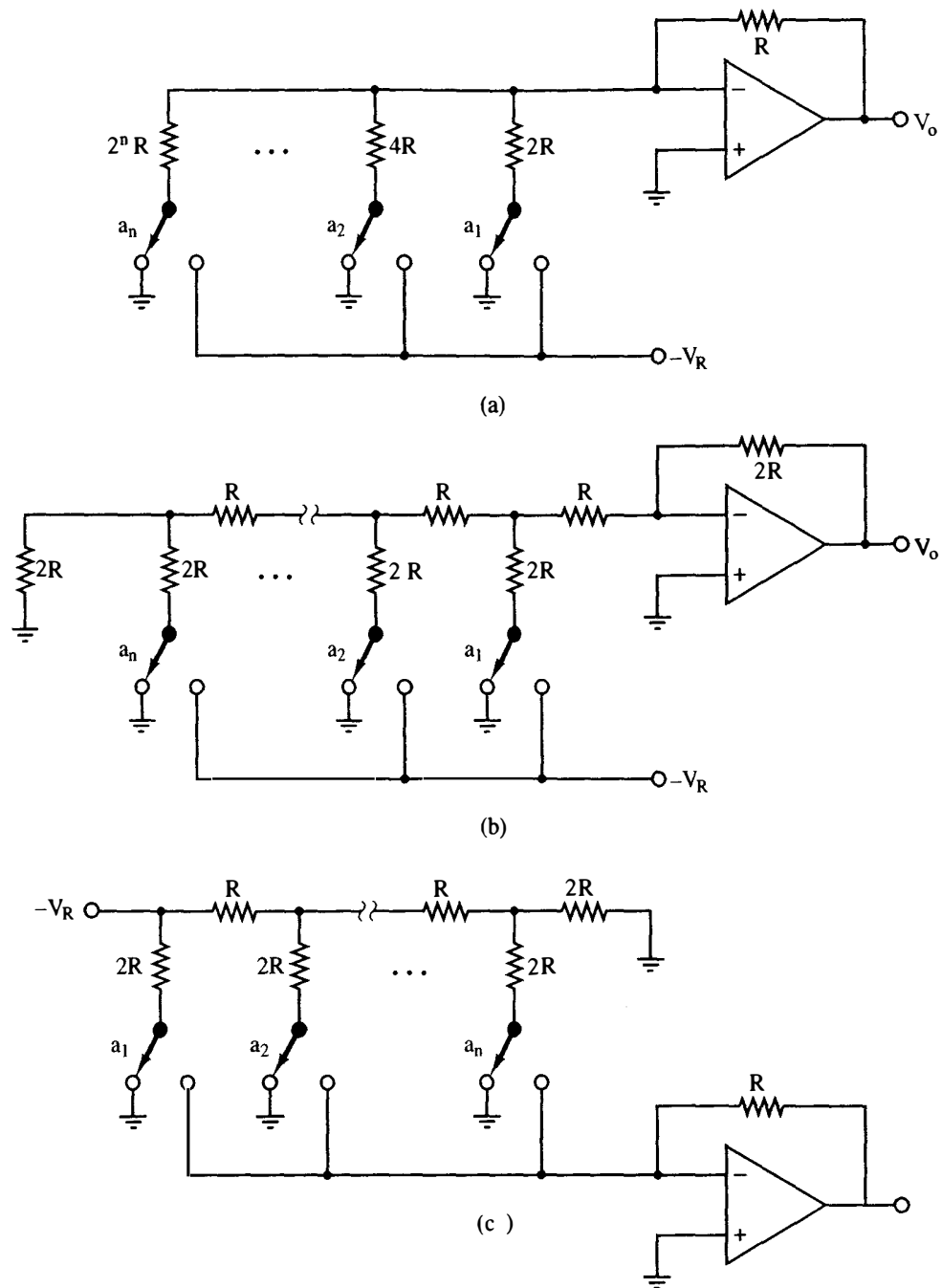


Figure 3-20 Digital-to-analog converters: (a) weighted resistor DAC; (b) R - $2R$ ladder; (c) inverted R - $2R$ ladder. [From R. C. Jaeger, "Tutorial: Analog Data Acquisition Technology, Part I—Digital-to-Analog Conversion," *IEEE MICRO*, Vol. 2, No. 2, May 1982: (a) Fig. 9, p. 25, (b) Fig. 11, p. 26, (c) Fig. 12, p. 28. © 1982 IEEE.]

of resistor values. It is well suited for integrated circuits since it requires only two resistor values R and $2R$. The value of R typically ranges from 2.5 to 10 k Ω . Taking successive Thévenin equivalent circuits [7] for each bit of the ladder, it is easy to show that the inputs are each reduced by a factor of 2 going from the MSB to the LSB.

Again, this network is using the switches in a voltage switching mode and requires low on-resistance, zero offset voltage switches.

Because the currents flowing in the ladder change as the input word changes, power dissipation and heating in the network change, causing nonlinearity in the DAC. Also, the load on the reference voltage depends on the binary input. Because of this, most monolithic versions of this DAC use the configuration shown in Figure 3-20c, known as the inverted R - $2R$ ladder. Here the currents flowing in the ladder are constant with the digital input diverting the current either to ground or to the input of a current-to-voltage converter. This is a popular configuration used with the CMOS process, which provides excellent switching devices. The switches still need to be low-on-resistance devices to minimize errors within the converter. The R - $2R$ ladder must be diffused, implanted, or thin-film, depending on the manufacturer's processing capability and the resolution of the DAC, and is used in DACs of up to 12 bits resolution.

Bipolar transistors do not perform well as voltage switches because of their inherent offset voltage in the saturated region of operation. However, they do make excellent current switches, and most DACs realized using bipolar processes use some form of switched current sources.

3.9 ANALOG-TO-DIGITAL CONVERSION*

The basic conversion scheme for most analog-to-digital conversion is shown in Figure 3-21a. The unknown voltage is connected to one input of an analog signal comparator and a time-dependent reference voltage is connected to the second input.

The transfer characteristic of the comparator is shown in Figure 3-21b. If the input voltage V_1 is greater than V_2 , the output voltage will be at a positive level corresponding to a logic "1." If input V_2 is greater than V_1 , the output voltage will be at a low level, corresponding to a logic "0."

To perform a conversion, the reference voltage V_R is varied to determine which of the 2^n possible binary words is closest to the unknown voltage V_x . The reference voltage V_R can assume 2^n different values of the form

$$V_R = V_r \sum_{i=1}^n A_i 2^{-i} \quad (3-41)$$

where V_r is a dc reference voltage and A_i are binary coefficients. The logic of the A/D converter attempts to choose the coefficients A_i so that the difference between the unknown input V_x and V_R is a minimum; that is, choose the A_i such that

$$\text{error} = |V_x - V_R| = |V_x - V_r \sum_{i=1}^n A_i 2^{-i}| \quad (3-42)$$

*Section 3.9, contributed by Richard C. Jaeger, is based on his "Data Acquisition Systems," Tutorial Notes, IECI 81, San Francisco, Nov. 9, 1981, and is included here with his permission.

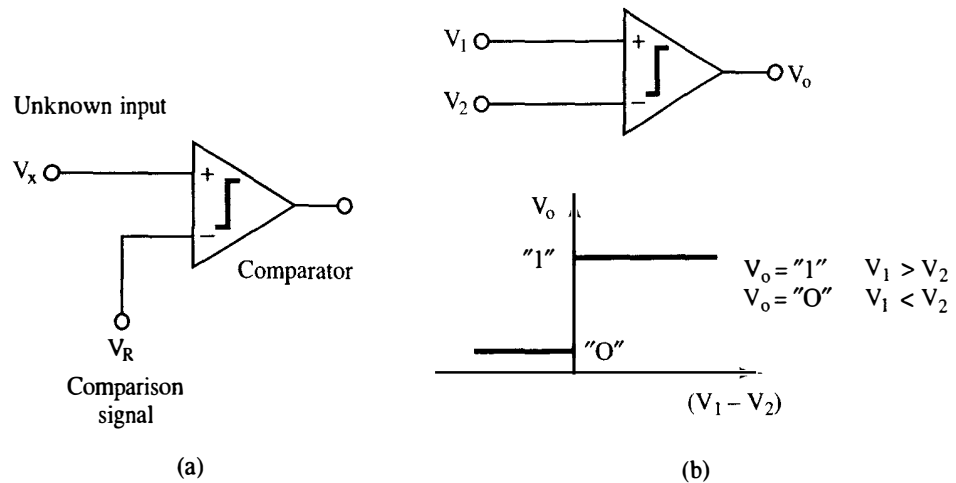


Figure 3-21 Analog-to-digital conversion: (a) general scheme; (b) comparator. [From R. C. Jaeger, "Tutorial: Analog Data Acquisition Technology, Part II—Analog-to-Digital Conversion," *IEEE MICRO*, Vol. 2, No. 3, Aug. 1982: (a) Fig. 5, p. 48, (b) Fig. 6, p. 48. © 1982 IEEE.]

is minimum. The basic difference in converters is in the strategy that is used to vary V_R to determine the binary coefficients A_i .

Counter Ramp Converter

One of the simplest ways of generating the comparison voltage V_R uses a D/A converter (DAC). An n -bit DAC can be used to generate any one of its possible 2^n outputs by simply applying the appropriate digital inputs. See (3-40) with $V_{fs} = V_r$. The most direct way to determine the unknown voltage V_x is to sequentially compare it to each possible DAC output. By connecting the input of the D/A converter to an n -bit binary counter, a step-by-step comparison with the unknown input can be made, as shown in Figure 3-22. The output of the D/A converter looks like a staircase during the conversion. A reset pulse (start of conversion, SOC) sets the counter output at zero. Each successive clock pulse increments the counter until the output of the DAC is larger than the unknown input V_x . At this point the comparator switches state and prevents any further clock pulses from incrementing the counter. The change of state of the comparator output indicates that the conversion cycle is complete and the contents of the binary counter represent the converted value of the input signal.

Several features of this converter should be noted. First, the length of the converter cycle is variable and proportional to the unknown input V_x . The maximum conversion period T_c occurs for a full-scale input signal corresponding to 2^n clock periods, or $T_c = 2^n/f_c$, where f_c is the clock pulse frequency. Second, the binary value in the counter represents the smallest DAC output voltage which is larger than the unknown input and is not necessarily the DAC output which is closest to the unknown voltage as we had originally hoped. Finally, if the unknown input should

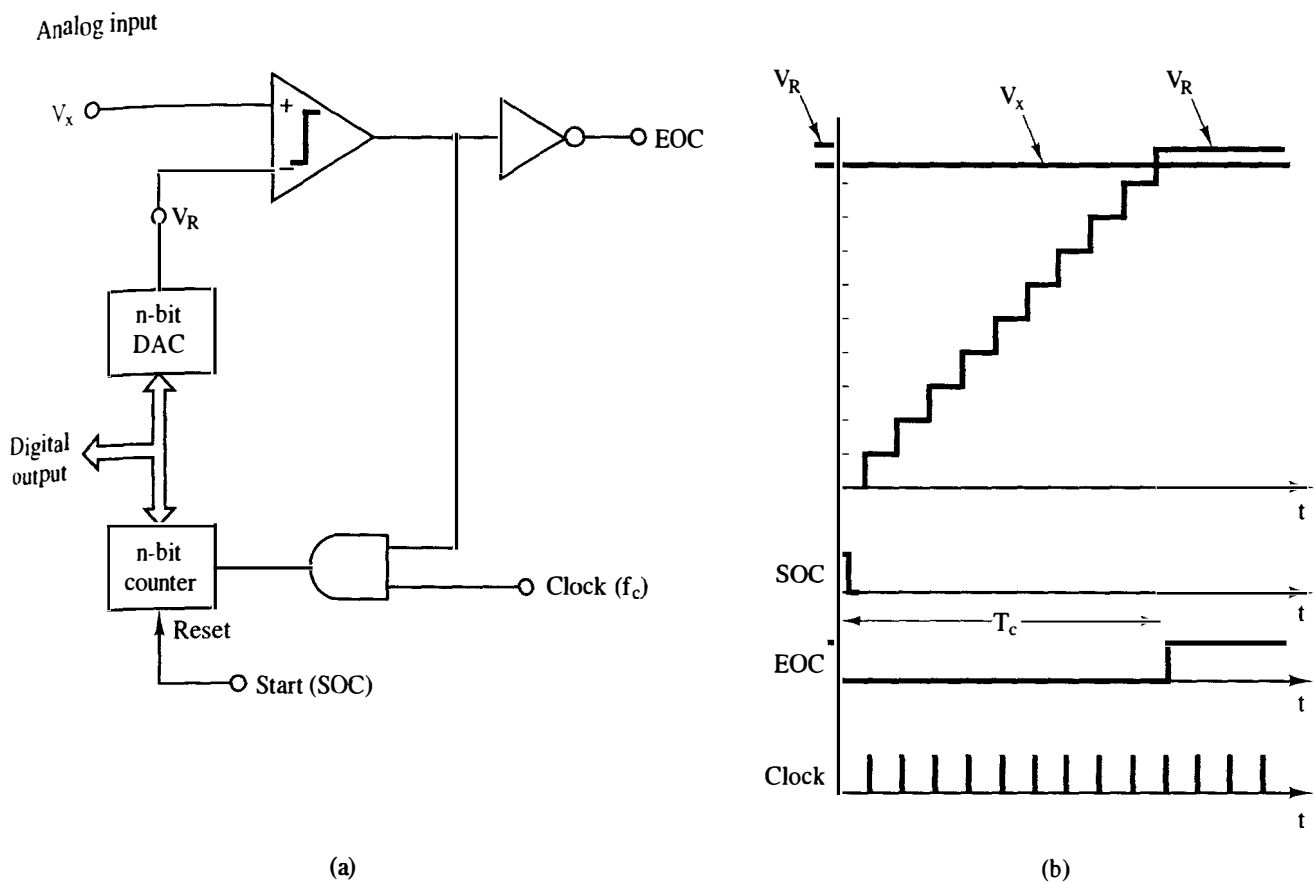


Figure 3-22 Counter ramp ADC: (a) block diagram; (b) timing diagram.

decrease or increase during the conversion period T_c , the binary output of the converter will be a true representation of the input voltage at the instant the comparator stops the counter.

The advantage of this type of converter is that it requires a small amount of hardware and is inexpensive to implement. For this reason, some of the least expensive monolithic and modular converters use this method. The main disadvantage is the relatively low conversion rate for a given converter. An n -bit converter requires 2^n clock periods for its longest conversion.

Tracking ADC

One attempt to improve the counting converter's performance is to modify it to track the input signal (see Figure 3-23). An up-down counter is used with slightly more complicated logic to force the output of the DAC to track changes in the unknown input V_x . If the comparator output indicates that the DAC output is less than V_x , the next clock pulse increments the counter. If the DAC output is greater than V_x , the next clock pulse decrements the counter. A staircase DAC output occurs until the converter "acquires" the input signal. If the unknown input is constant, the DAC output will alternate between two output values which differ by one LSB. If the

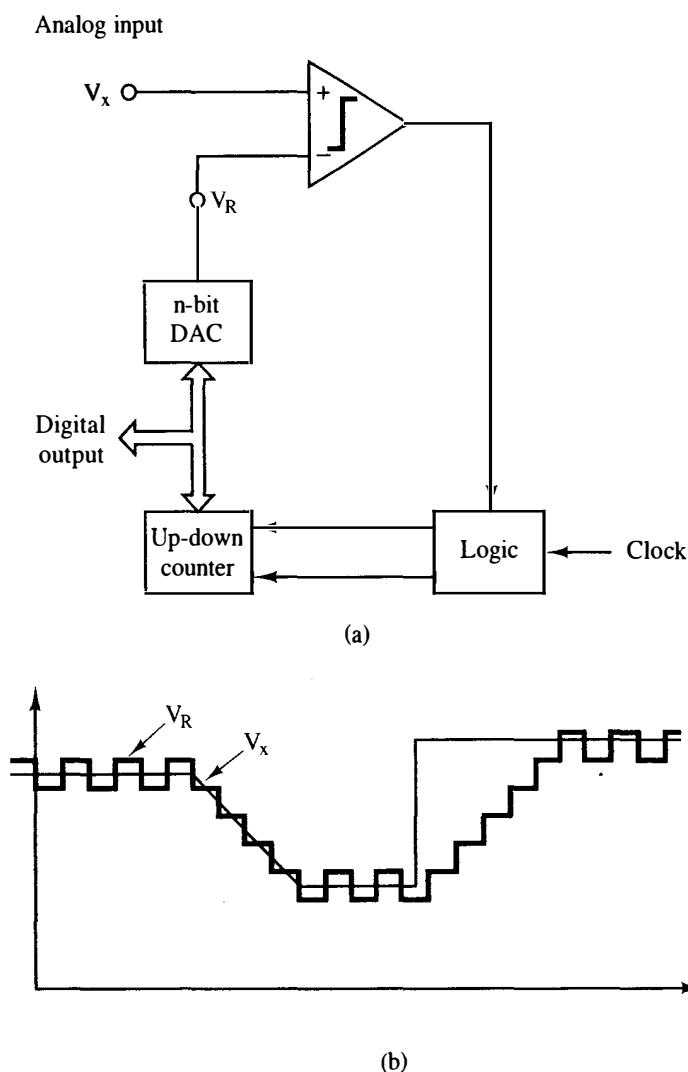


Figure 3-23 Tracking ADC: (a) block diagram; (b) example waveforms. (From R. C. Jaeger, "Tutorial: Analog Data Acquisition Technology, Part II—Analog-to-Digital Conversion," *IEEE MICRO*, Vol. 2, No. 3, Aug. 1982, Fig. 8, p. 50, © 1982 IEEE.)

unknown input changes slowly enough, the DAC output will follow the unknown input. Thus the counter contents always contain an accurate representation of the input signal at the time of the last clock pulse, and the converted value may be read from this counter at any time. However, if V_x changes too rapidly, the converter will not be able to follow quickly enough and the counter contents no longer correctly represent the output voltage. Also, when a new input signal is applied, this converter takes exactly the same amount of time to acquire the signal as the counter converter takes to reach conversion. Because of their relatively low conversion rate for a given clock frequency, other types of converters are usually used when the input signal is expected to change by large amounts from one conversion time to the next.

Successive-Approximation ADC

The successive-approximation converter uses a different strategy in varying the applied reference input to the comparator and results in a converter that requires only n clock periods to complete an n -bit conversion.

The operation of a 3-bit converter is shown in Figure 3-24. A *binary search* is used to determine the best approximation to V_x . After reset, the successive approximation logic (SAL) sets the DAC output to $0.5V_{fs}$ and checks the comparator output. At the next clock pulse, the DAC output is set to $0.75V_{fs}$ if the DAC output was less than V_x and to $0.25V_{fs}$ if the DAC output had been greater than V_x . Again the comparator output is tested, and the next clock pulse causes the DAC output to be incremented or decremented by $V_{fs}/8$. A third comparison is made. The final converted binary output is not changed if V_x was larger than the DAC output or is decremented by one LSB if V_x was less than the DAC output. Thus the conversion is obtained at the end of three clock periods for the 3-bit converter or n clock periods for an n -bit converter. The 3-bit DAC code sequence is illustrated in Figure 3-25.

Fast conversion rates are possible with this converter, and it is a very popular type used for 8- to 16-bit converters. The primary factors limiting the speed of this

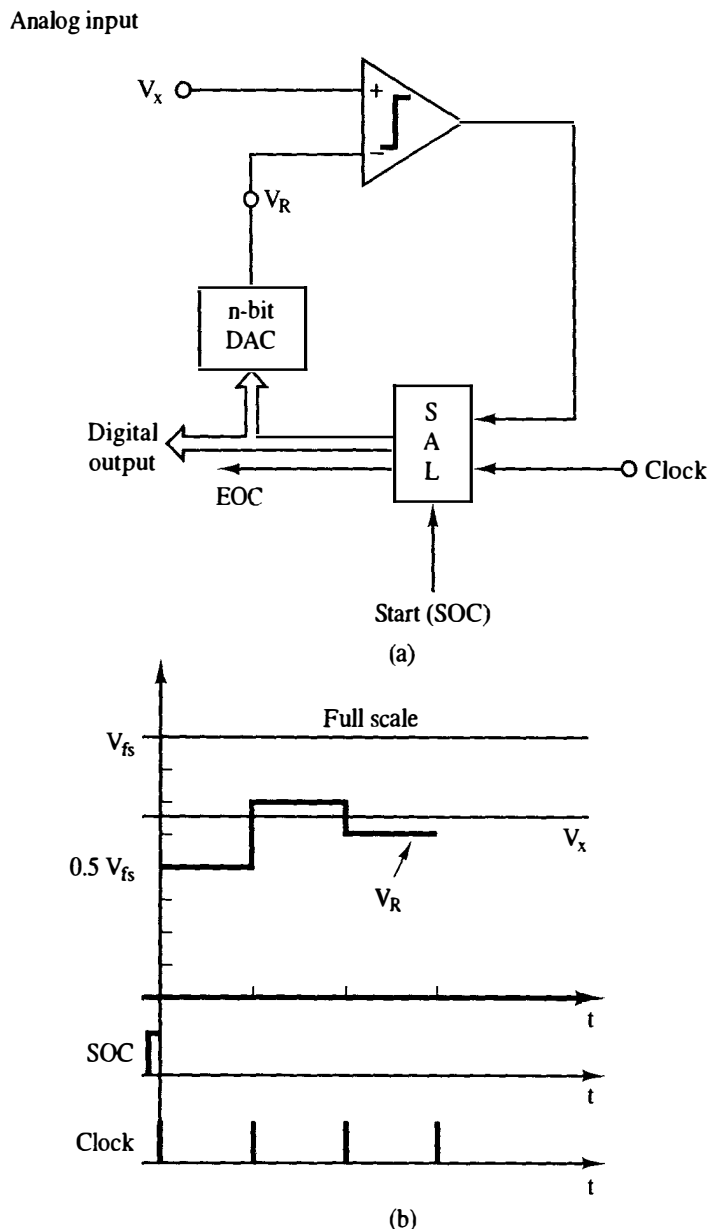


Figure 3-24 Successive-approximation ADC: (a) block diagram; (b) timing diagram. (From R. C. Jaeger, "Tutorial: Analog Data Acquisition Technology, Part II—Analog-to-Digital Conversion," *IEEE MICRO*, Vol. 2, No. 3, Aug. 1982, Fig. 9, p. 50. © 1982 IEEE.)

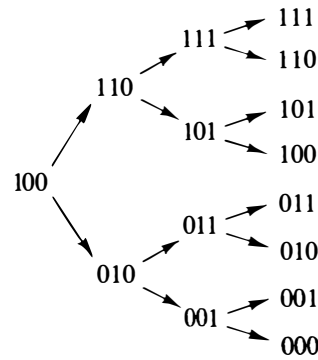


Figure 3-25 Three-bit successive-approximation DAC code sequence.

ADC are the time required for the D/A converter output to settle within a fraction of a $V_{fs}/2^n$ (i.e., LSB) and the time for the comparator to respond to input signals which may differ by very small amounts.

A problem in the application of these converters is that the input must remain constant during the full conversion period. Otherwise, the digital output of the converter does not bear any precise relation to the value of the unknown input voltage V_x . Sample-and-hold circuits are usually used ahead of the successive-approximation ADC to avoid this problem [8].

Single-Ramp Converter

The discrete output of the D/A converter in the counter ramp ADC may be replaced with a continuously increasing reference signal as shown in Figure 3-26. In this case the reference is usually called a ramp and varies from slightly below zero to V_{fs} . The length of time required for the ramp signal to become equal to the unknown voltage is then proportional to the unknown voltage. This time period T_c is quantized with a counter to produce the binary representation of the unknown input signal. Referring to the figure, converter operation begins with a start of conversion (SOC) signal, which resets the binary counter and starts the ramp generator at a slightly negative value. As the ramp output crosses through zero, the output of comparator 2 enables the AND GATE, allowing clock pulses to accumulate in the counter. The counter continues counting until the ramp output equals the unknown voltage V_x . At this time comparator 1 disables the AND GATE, which stops the counter. The number of clock pulses in the counter is directly proportional to the input voltage since $V_x = KT_c$, where K is the slope of the ramp in volts per second and $T_c = N/f_c$. If the slope of the ramp is chosen to be $V_{fs}f_c/2^n$, the unknown voltage is given by

$$V_x = \frac{V_{fs} N}{2^n}$$

and

$$\frac{V_x}{V_{fs}} = \frac{N}{2^n}$$

The number N in the counter can be directly interpreted as the binary fraction equivalent to V_x/V_{fs} .

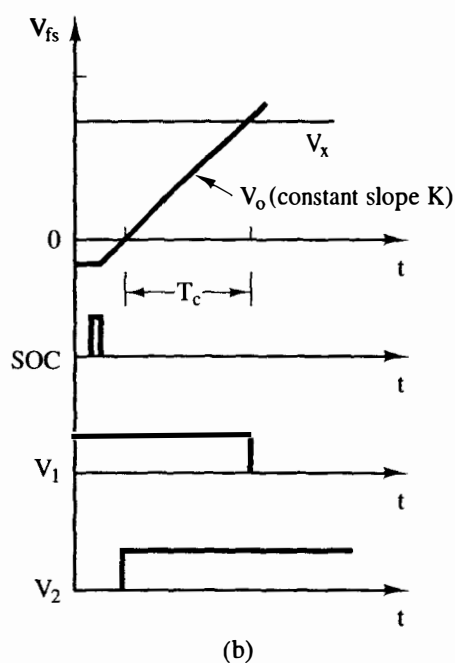
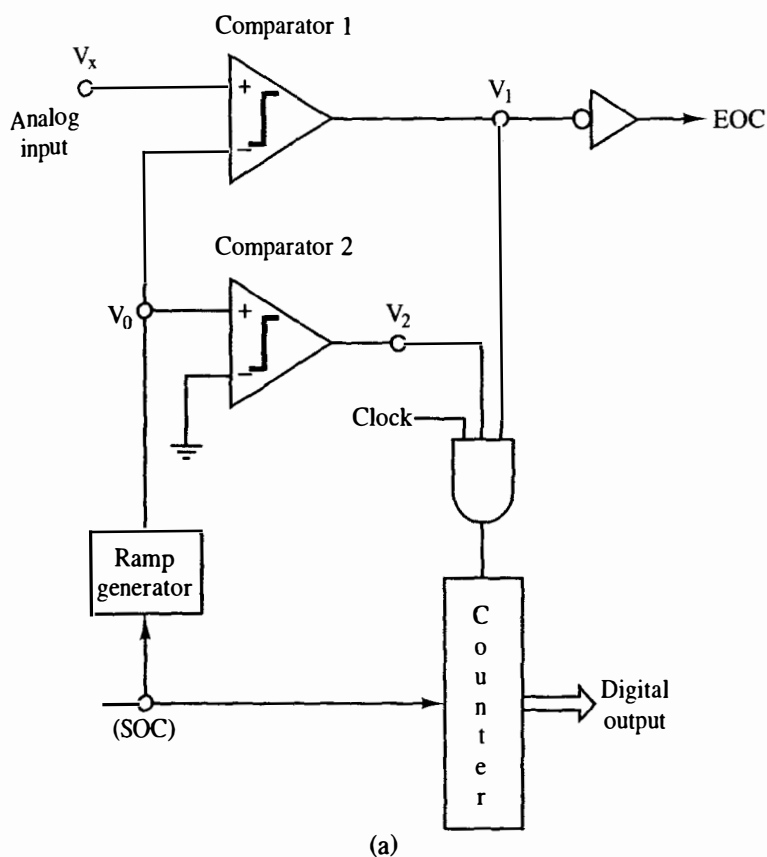


Figure 3-26 Single-ramp ADC: (a) block diagram ($V_x = KT_c$; N , counter output); (b) timing diagrams.

The conversion time T_c of the single-ramp converter is clearly variable and proportional to the unknown voltage V_x . Maximum conversion time occurs for $V_x = V_{fs}$, or $T_c = 2^n/f_c$. As in the case for the counter converter, the digital output represents the value of V_x at the instant the end of conversion signal occurs.

The ramp voltage is usually generated using an integrator connected to a

constant reference voltage as shown in Figure 3-27 [6]. When the reset switch is opened, the output increases with a constant slope as given by

$$V_o = -\frac{1}{RC} \int_0^{T_c} V_r dt = -\frac{V_r}{RC} t \Big|_0^{T_c} \quad (3-43)$$

For a constant reference $-V_r$, the slope of the ramp is V_r/RC , which points out one of the major limitations of the converter. The slope is dependent on the values of R and C , which are difficult to maintain constant in the presence of temperature variations and over long periods of time. Because of this problem, the single-ramp converter is seldom used today.

Dual-Ramp Converter

The dual-ramp converter solves the problem associated with the single-ramp converter and is a common converter found in data acquisition and control instrumentation systems. Converter operation may be understood by referring to Figure 3-28. The converter operates with a conversion cycle consisting of two separate integration intervals. First, the unknown V_x is integrated for a known period of time. Then the value of this integral is compared to that of a known reference which is integrated for a variable length of time.

At the start of conversion, the counter is reset, and the integrator is reset to a slightly negative voltage. The unknown input V_x is connected to the integrator input through switch S_1 . V_x is integrated for a fixed period of time $T_1 = 2^n/f_c$ which begins when the integrator output crosses through zero. At the end of time T_1 the counter overflows, which causes S_1 to be turned off and the reference input $+V_r$ to be connected to the integrator input through S_2 . The integrator output decreases until it crosses through zero, and the comparator changes state, indicating the end of conversion. The number in the counter represents the converted value of the unknown V_x , as demonstrated below.

Circuit operation forces the integrals over the time periods $0+$ to T_1 and T_1 to $(T_1 + T_2)$ to be equal, so that

$$\frac{1}{RC} \int_0^{T_1} V_x dt = \frac{1}{RC} \int_{T_1}^{T_1 + T_2} V_r dt \quad \text{or} \quad \frac{\bar{V}_x T_1}{RC} = \frac{V_R T_2}{RC} \quad (3-44)$$

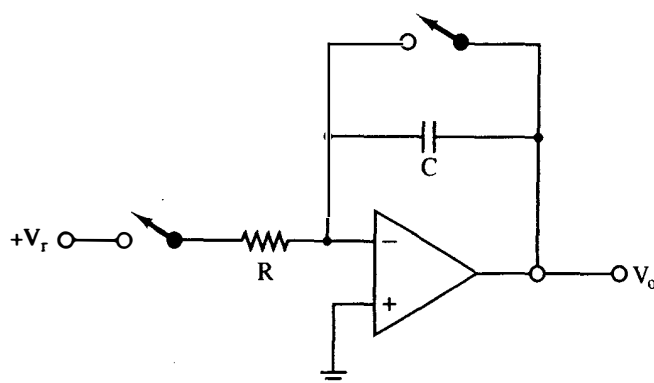


Figure 3-27 Ramp generator.

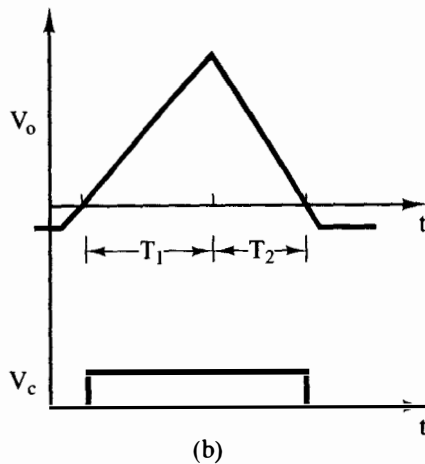
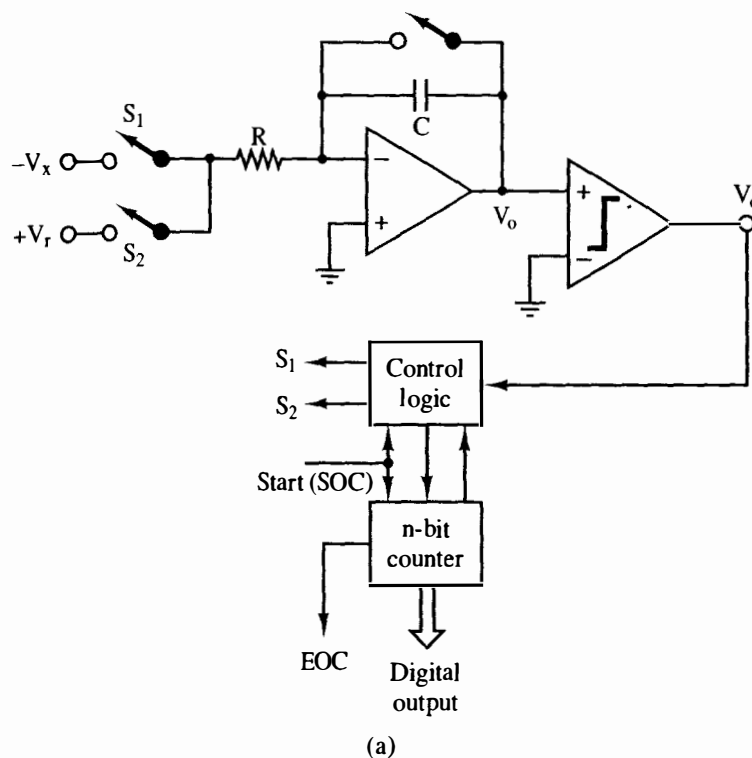


Figure 3-28 Dual-ramp ADC: (a) block diagram; (b) control waveforms. (From R. C. Jaeger, "Tutorial: Analog Data Acquisition Technology, Part II—Analog-to-Digital Conversion," *IEEE MICRO*, Vol. 2, No. 3, Aug. 1982, Fig. 13, p. 53. © 1982 IEEE.)

T_1 is set equal to $2^n/f_c$ since the unknown V_x was integrated for the amount of time needed for the n -bit counter to overflow. \bar{V}_x is the average of V_x . The time period T_2 is equal to N/f_c where N is the number of counts accumulated in the counter during the second phase of operation. The average value of the input is then given by

$$\bar{V}_x = \frac{V_r N}{2^n} \quad (3-45)$$

assuming that the RC product remained constant throughout the complete conversion cycle. The values of R and C no longer enter directly into the relation between V_x and V_r , and the stability problem associated with the single-ramp converter has been overcome. Furthermore, the digital output word represents the average value

of V_x during the first integration phase. Thus V_x is allowed to change during the conversion cycle of this converter without destroying the validity of the converted output.

The conversion time T_c requires 2^n clock periods for the first integration period and N clock periods for the second integration interval. Thus the conversion time is variable and

$$T_c = \frac{N + 2^n}{f_c} \quad (3-46)$$

As mentioned above, the binary output of the dual-ramp converter represents the average of the input during the first integration phase. The integrator operates as a low-pass filter. Any sinusoidal input signals whose frequencies are exact multiples of the reciprocal of the integration time T_1 will have integrals of zero value and will not disturb the converter output. This property is often used in digital voltmeters which use dual-ramp converters with an integration time which is some fixed multiple of the period of the 50- or 60-Hz power source. Noise sources at multiples of the power-line frequency are removed by the integrating ADC. This property is usually called *normal mode rejection*.

The dual ramp is a widely used converter. Its integrating properties combined with careful design allow accurate conversion at resolutions exceeding 20 bits with low conversion speeds. The basic dual ramp has been modified to include automatic offset elimination phases in a number of monolithic converters.

Parallel Converter (Flash)

The fastest converters use additional hardware to perform a parallel rather than serial conversion. Figure 3-29 schematically shows a 3-bit parallel converter in which the unknown input V_x is simultaneously compared to seven different reference values. The logic network converts the comparator outputs directly to the 3-bit digital value that corresponds to the input. The speed of the converter can be very fast since the conversion speed is limited only by the speed of the comparators and of the logic network. Also, the output continuously represents the input except for the comparator and logic delays. Thus the converter can be thought of as automatically tracking the input signal.

This type of converter is used when maximum speed is needed, and is usually found in relatively low resolution converters since $2^n - 1$ comparators and reference voltages are required for an n -bit converter. Thus the cost of implementing such a converter grows rapidly with resolution. The term *flash* is sometimes used as the name of the parallel converter because of its inherent speed.

ADC Comparison

A comparison of ADC characteristics is displayed in Table 3-1. The relative conversion rate and complexity of each scheme is considered. Here we note that speed and

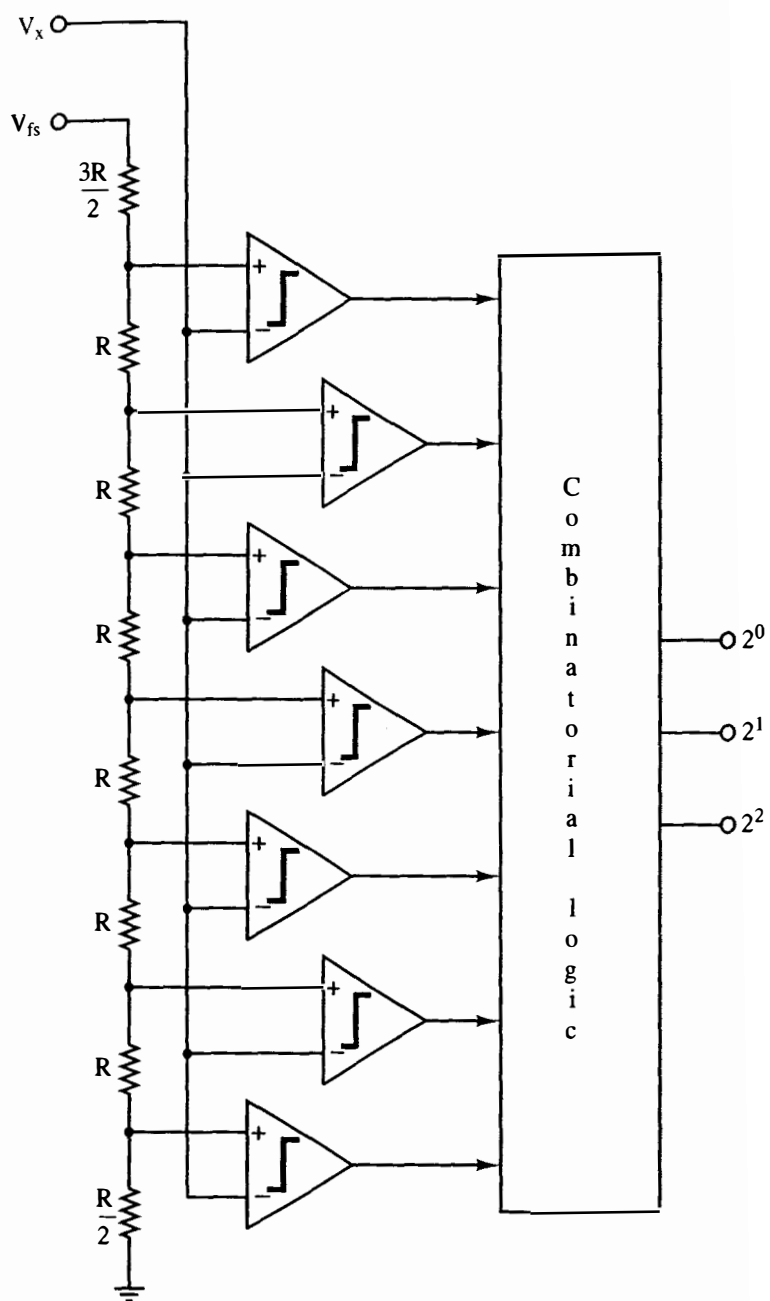


Figure 3-29 Parallel, or "flash" ADC.

TABLE 3-1 ADC COMPARISON

Converter type	Conversion rate ^a	Cost/complexity	Comments
Tracking		Low	Needs slowly varying input signal; output always available
Counter ramp	Slow	Low	Needs stable input
Single ramp	Slow	Low	Lacks stability with time and temperature
Dual ramp	Slow	Medium	Integrates input signal; can be used at high resolution—20 bits or more
Successive approximation	Fast	Medium	Needs stable input
Parallel “flash”	Fastest	High	Output always available

^aRamp converters have variable conversion time; successive approximation and parallel converters have fixed conversion time.

cost are directly related. The medium-speed, medium-cost successive approximation ADC is a common choice in digital control systems.

3.10 SUMMARY

The material presented in this chapter is fundamental to the study of sampled-data control systems. The sampling and reconstruction processes are an integral part of these systems, and it is important to have a good understanding of these topics before proceeding to the more advanced topics in analysis and design. The mathematical model of the sample-hold operation, Figure 3-5, is basic to all analysis and design of digital control systems.

An important result of the topics developed in this chapter is the development of approximate rules for the choice of the sample period T for a given signal. Whereas this chapter emphasized the importance of the frequency contents of a signal in determining its sampling rate, later chapters will show the importance of a system's frequency response in determining the sample rate to be used in that system.

A survey of digital-to-analog and analog-to-digital conversion methods was included in this chapter to provide some practical information with which to select D/A and A/D converters for particular applications.

REFERENCES AND FURTHER READING

1. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2d ed. Englewood Cliffs, NJ: Prentice Hall, 1991.
2. F. G. Stremmer, *Introduction to Communication Systems*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1982.
3. G. Doetsch, *Guide to the Applications of the Laplace and z-Transforms*. New York: Van Nostrand Reinhold, 1971.
4. C. L. Phillips and J. M. Parr, *Signals, Systems, and Transforms*. Englewood Cliffs, NJ: Prentice Hall, 1994.
5. R. M. Oliver, J. R. Pierce, and C. E. Shannon, "The Philosophy of Pulse Code Modulation," *Proc. IRE*, Vol. 36, No. 11, pp. 1324–1331, Nov. 1948.
6. S. Wolf, *Guide to Electronic Measurements and Laboratory Practice*. Englewood Cliffs, NJ: Prentice Hall, 1983.
7. M. E. Van Valkenburg, *Network Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1974.
8. Analog Devices, *Analog-Digital Conversion Handbook*. Englewood Cliffs, NJ: Prentice Hall, 1986.
9. R. V. Churchill, *Complex Variables and Applications*. New York: McGraw-Hill Book Company, 1974.
10. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1988.
11. B. C. Kuo, *Digital Control Systems*, 2d ed. New York: Saunders College Publishing, 1992.
12. C. L. Phillips, D. L. Chenoweth, and R. K. Cavin III, "z-Transform Analysis of Sampled-Data Control Systems without Reference to Impulse Functions," *IEEE Trans. Educ.*, Vol. E-11, pp. 141–144, June 1968.
13. C. R. Wylie, Jr., *Advanced Engineering Mathematics*, 4th ed. New York: McGraw-Hill Book Company, 1975.

PROBLEMS

- 3-1. (a) Give the definition of the starred transform.
(b) Give the definition of the z-transform.
(c) For a function $e(t)$, derive a relationship between its starred transform $E^*(s)$ and its z-transform $E(z)$.
- 3-2. A signal $e(t)$ is sampled by the ideal sampler as specified by (3-3).
(a) List the conditions under which $e(t)$ can be *completely* recovered from $e^*(t)$, that is, the conditions under which *no* loss of information by the sampling process occurs.
(b) State which of the conditions listed in part (a) can occur in a physical system. Recall that the ideal sampler itself is not physically realizable.
(c) Considering the answers in part (b), state why we can successfully employ systems that use sampling.

- 3-3. A system is defined as linear if the principle of superposition applies. Is a sampler/zero-order-hold device linear? Prove your answer.
- 3-4. Use the residue method of (3-10) to find the starred transform of the following functions.

$$(a) E(s) = \frac{20}{(s+2)(s+5)}$$

$$(b) E(s) = \frac{5}{s(s+1)}$$

$$(c) E(s) = \frac{s+2}{s(s+1)}$$

$$(d) E(s) = \frac{s+2}{s^2(s+1)}$$

$$(e) E(s) = \frac{s^2 + 5s + 6}{s(s+4)(s+5)}$$

$$(f) E(s) = \frac{2}{s^2 + 2s + 5}$$

- 3-5. Find $E^*(s)$ for each of the following functions. Express $E^*(s)$ in closed form.

$$(a) e(t) = e^{at}$$

$$(b) E(s) = \frac{e^{-2Ts}}{s-a}$$

$$(c) e(t) = e^{a(t-2T)} u(t-2T)$$

$$(d) e(t) = e^{a(t-T/2)} u(t-T/2)$$

- 3-6. For $e(t) = e^{-3t}$:

(a) Express $E^*(s)$ as a series.

(b) Express $E^*(s)$ in closed form.

(c) Express $E^*(s)$ as a series which is different from that in part (a).

- 3-7. Express the starred transform of $e(t-kT)u(t-kT)$, k an integer, in terms of $E^*(s)$, the starred transform of $e(t)$. Base your derivation on (3-3).

- 3-8. Find $E^*(s)$ for

$$E(s) = \frac{1 - e^{-Ts}}{s(s+1)}$$

- 3-9. (a) Find $E^*(s)$, for $T = 0.1$ s, for the two functions below. Explain why the two transforms are equal, first from a time-function approach, and then from a pole-zero approach.

$$(i) e_1(t) = \cos(4\pi t)$$

$$(ii) e_2(t) = \cos(16\pi t)$$

(b) Give a third time function that has the same $E^*(s)$.

- 3-10. Compare the pole-zero locations of $E^*(s)$ in the s -plane with those of $E(s)$, for the function given in Problem 3-4(c).

- 3-11. Find $E^*(s)$, with $T = 0.5$ s, for

$$E(s) = \frac{(1 - e^{-0.5s})^2}{0.5s^2(s+1)}$$

- 3-12. Suppose that $E^*(s) = [e(t)]^* = 1$.

(a) Find $e(kT)$ for all k .

(b) Can $e(t)$ be found? Justify your answer.

(c) Sketch two different continuous-time functions that satisfy part (a).

(d) Write the equations for the two functions in part (c).

- 3-13. Suppose that the signal $e(t) = \cos[(\omega_s/2)t + \theta]$ is applied to an ideal sampler and zero-order hold.

- (a) Show that the amplitude of the time function out of the zero-order hold is a function of the phase angle θ by sketching this time function.
 - (b) Show that the component of the signal out of the data hold, at the frequency $\omega = \omega_s/2$, is a function of the phase angle θ by finding the Fourier series for the signal.
- 3-14.** (a) A sinusoid with a frequency of 2 Hz is applied to a sampler/zero-order hold combination. The sampling rate is 10 Hz. List all the frequencies present in the output that are less than 50 Hz.
- (b) Repeat part (a) if the input sinusoid has a frequency of 8 Hz.
- (c) The results of parts (a) and (b) are identical. Give three other frequencies, which are greater than 50 Hz, that yield the same results as parts (a) and (b).
- 3-15.** Given the signal $e(t) = 3 \sin 4t + 2 \sin 7t$.
- (a) List all frequencies less than $\omega = 50$ rad/s that are present in $e(t)$.
- (b) The signal $e(t)$ is sampled at the frequency $\omega_s = 22$ rad/s. List all frequencies present in $e^*(t)$ that are less than $\omega = 50$ rad/s.
- (c) The signal $e^*(t)$ is applied to a zero-order hold. List all frequencies present in the hold output that are less than $\omega = 50$ rad/s.
- (d) Repeat part (c) for $e^*(t)$ applied to a first-order hold.
- 3-16.** A signal $e(t) = 4 \sin 7t$ is applied to a sampler/zero-order-hold device, with $\omega_s = 4$ rad/s.
- (a) What is the frequency component in the output that has the largest amplitude?
- (b) Find the amplitude and phase of that component.
- (c) Sketch the input signal and the component of part (b) versus time.
- (d) Find the ratio of the amplitude in part (b) to that of the frequency component in the output at $\omega = 7$ rad/s (the input frequency).
- 3-17.** It is well known that the addition of phase lag to a closed-loop system is destabilizing. A sampler/data-hold device adds phase lag to a system, as described in Section 3.7. A certain analog control system has a bandwidth of 10 Hz. By this statement we mean that the system (approximately) will respond to frequencies less than 10 Hz, and (approximately) will not respond to frequencies greater than 10 Hz. A sampler/zero-order-hold device is to be added to this control system.
- (a) It has been determined that the system can tolerate the addition of a maximum of 10° phase lag within the system bandwidth. Determine the approximate minimum sampling rate allowed, along with the sample period T .
- (b) Repeat part (a) for a maximum of 5° phase lag.
- (c) Repeat part (a) for a maximum of 20° phase lag.
- 3-18.** A sinusoid is applied to a sampler/zero-order-hold device, with a distorted sine wave appearing at the output, as shown in Figure 3-15.
- (a) With the sinusoid of unity amplitude and frequency 2 Hz, and with $f_s = 12$ Hz, find the amplitude and phase of the component in the output at $f_1 = 2$ Hz.
- (b) Repeat part (a) for the component in the output at $(f_s - f_1) = 10$ Hz.
- (c) Repeat parts (a) and (b) for a sampler-first-order-hold device.
- (d) Comment on the distortion in the data-hold output for the cases considered in parts (a), (b), and (c).

- 3-19. A polygonal data hold is a device that reconstructs the sampled signal by the straight-line approximation shown in Figure P3-19. Show that the transfer function of this data hold is

$$G(s) = \frac{\epsilon^{Ts}(1 - \epsilon^{-Ts})^2}{Ts^2}$$

Is this data hold physically realizable?

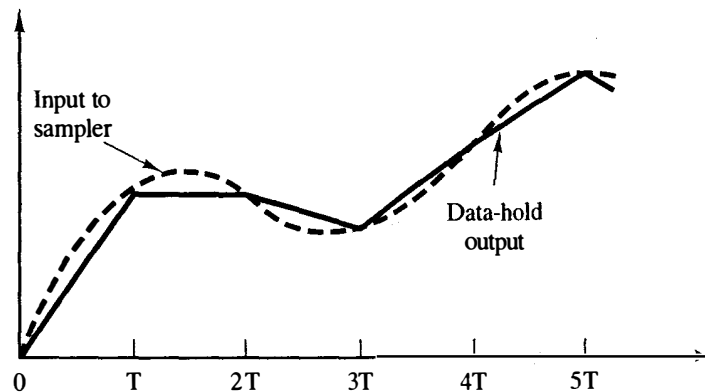


Figure P3-19 Response of a polygonal data hold.

- 3-20. A data hold is to be constructed that reconstructs the sampled signal by the straight-line approximation shown in Figure P3-20. Note that this device is a polygonal data hold (see Problem 3-19) with a delay of T seconds. Derive the transfer function for this data hold. Is this data hold physically realizable?

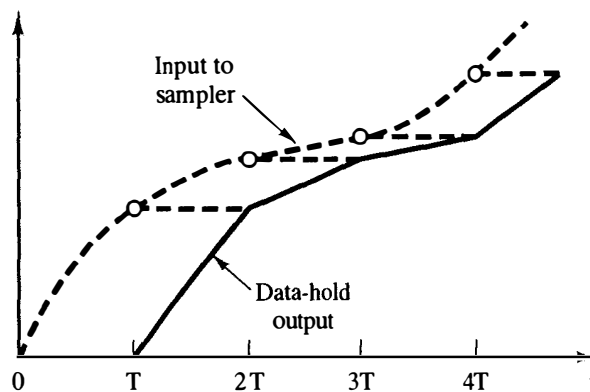


Figure P3-20 Response of a polygonal data hold with time delay.

- 3-21. Plot the ratio of the frequency responses (in decibels) and phase versus ω for the data holds of problems 3-19 and 3-20. Note the effect on phase of making the data hold realizable.
- 3-22. Derive the transfer function of the fractional-order hold [see (3-39)].
- 3-23. Shown in Figure P3-23 is the output of a data hold that clamps the output to the input for the first half of the sampling period, and returns the output to a value of zero for the last half of the sampling period.
- (a) Find the transfer function of this data hold.

- (b) Plot the frequency response of this data hold.
 (c) Comparing this frequency response to that of the zero-order hold, comment on which would be better for data reconstruction.

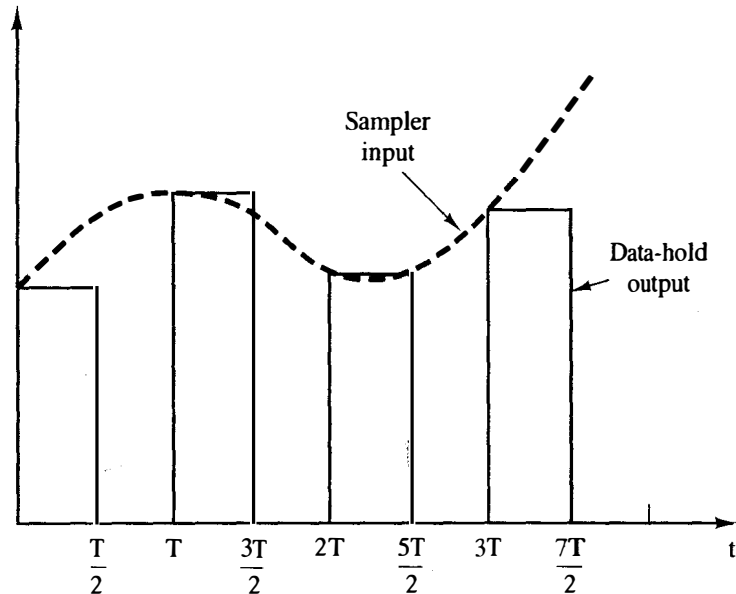


Figure P3-23 Data hold output for Problem 3-23.

- 3-24.** Consider a 0–10 V, 4-bit digital-to-analog converter. By this specification we mean that, in (3-40), $V_R = 10$ V and $n = 4$.
 (a) Find the maximum possible output voltage.
 (b) Find the minimum possible nonzero change in the output voltage.
 (c) Consider a 0–10 V, 3-bit D/A. List all possible output voltages.
 (d) Consider a 0–10 V, n -bit D/A. Find n such that the minimum possible nonzero change in the output voltage is not greater than 0.005 V.
- 3-25.** Consider a 0- to 10-V, 8 bit analog-to-digital converter. By this specification, we mean that, in (3-41), $V_r = 10$ V and $n = 8$.
 (a) For an analog input voltage in the range 0 to 10 V, find the maximum error of conversion in (3-42).
 (b) Suppose that the converter is a counter ramp converter, and that the binary output is 00000100 [see (3-41)]. Find the range of the input voltage that can produce this binary output.
- 3-26.** Suppose that the clock used in a 12-bit A/D conversion is 5 MHz (see Figure 3-22).
 (a) Find the maximum conversion time, in seconds, for a counter ramp converter.
 (b) Repeat part (a) for a dual ramp converter.
 (c) Repeat part (a) for a successive approximation converter.
- 3-27.** Consider a 0–5 V, 4-bit successive approximation A/D. If the input analog voltage is 3.70 V, calculate the four successive voltages for V_R , as illustrated in Figure 3-24b for a 3-bit converter.
- 3-28.** Consider the system shown in Figure P3-28. For this problem there is no signal processing; that is, the 8 bits from the A/D are sent directly to the D/A. Find v_o if $v_i = 5.01$ V, and the A/D is a:

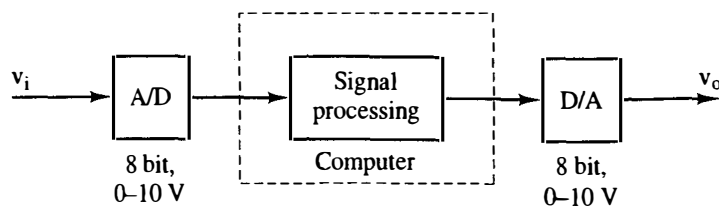


Figure P3-28 System for Problem 3-28.

- (a) Counter-ramp converter.
 - (b) Tracking converter.
 - (c) Successive-approximations converter.
 - (d) Dual-ramp converter.
 - (e) Parallel converter.
- 3-29.** For the system of Figure P3-28, suppose that the A/D is as shown, and the D/A is 12 bits, 0 to 10 V. For this problem, the code conversion appends 4 zero bits in the least-significant-bit positions to the 8 bits out of the A/D. These 12 bits are then sent to the D/A. Find v_o if $v_i = 5.01$ V and the A/D is a:
- (a) Counter-ramp converter.
 - (b) Tracking converter.
 - (c) Successive-approximations converter.
 - (d) Dual-ramp converter.
 - (e) Parallel converter.
- 3-30.** Repeat Problem 3-29 for the case that the code conversion appends 4 zero bits in the most-significant-bit positions to the 8 bits from the A/D.

Open-Loop Discrete-Time Systems

4.1 INTRODUCTION

In the preceding chapters, the topics of discrete-time systems, z -transforms, and sampling and data reconstruction were developed. In this chapter these developments are utilized to derive analysis methods for open-loop discrete-time systems. (The derivations are necessary, since the ideal sampler does not have a transfer function.) These analysis techniques will then be extended to closed-loop systems in the chapters that follow.

4.2 THE RELATIONSHIP BETWEEN $E(z)$ AND $E^*(s)$

To provide the proper background for our analysis of open-loop systems, let us establish the relationship that exists between $E(z)$ and $E^*(s)$. Recall that in Chapter 2 the z -transform of the number sequence $\{e(k)\}$ was defined in equation (2-7) to be

$$\mathcal{Z}[\{e(k)\}] = E(z) = e(0) + e(1)z^{-1} + e(2)z^{-2} + \cdots \quad (4-1)$$

In addition, the starred transform for the time function $e(t)$ was defined in equation (3-7) as

$$E^*(s) = e(0) + e(T)e^{-Ts} + e(2T)e^{-2Ts} + \cdots \quad (4-2)$$

The similarity between these two transforms is obvious. In fact, if we assume that the number sequence $\{e(k)\}$ is obtained from sampling a time function $e(t)$ [i.e., if $e(k)$ of (4-1) is equal to $e(kT)$ of (4-2)], and if $\epsilon^{sT} = z$ in (4-2), then (4-2) becomes the z -transform. Hence in this case

$$E(z) = E^*(s)|_{\epsilon^{sT} = z} \quad (4-3)$$

We see then that the z -transform can be considered to be a special case of the Laplace transform for our purposes. We will employ the change of variable in (4-3), and, in general, use the z -transform instead of the starred transform in our analysis of discrete-time systems. In addition, if the starred transform is required, we will first use the z -transform tables to find $E(z)$, and then use the inverse of (4-3) to find $E^*(s)$. This approach is illustrated by the following example.

Example 4.1

Let

$$E(s) = \frac{1}{(s+1)(s+2)}$$

Then, from the z -transform tables in Appendix VIII,

$$E(z) = E^*(s)|_{\epsilon^{sT} = z} = \frac{z(\epsilon^{-T} - \epsilon^{-2T})}{(z - \epsilon^{-T})(z - \epsilon^{-2T})}$$

Hence

$$E^*(s) = \frac{\epsilon^{Ts}(\epsilon^{-T} - \epsilon^{-2T})}{(\epsilon^{Ts} - \epsilon^{-T})(\epsilon^{Ts} - \epsilon^{-2T})}$$

which checks that derived in Example 3.3 via (3-10).

Note that in Example 4.1, $E^*(s)$ has an infinity of poles and zeros in the s -plane. However, $E(z)$ has only a single zero at $z = 0$, and two poles—one at ϵ^{-T} and the other at ϵ^{-2T} . Thus any analysis procedure that utilizes a pole-zero approach is greatly simplified through the use of the z -transform. Other advantages of this approach will become obvious as we develop analysis techniques for sampled-data systems.

$E(z)$ can now be calculated from (3-10) via the substitution in (4-3) as

$$E(z) = \sum_{\substack{\text{at poles} \\ \text{of } E(\lambda)}} \left[\text{residues of } E(\lambda) \frac{1}{1 - z^{-1} \epsilon^{T\lambda}} \right] \quad (4-4)$$

This expression is useful in generating z -transform tables. Because of the relation between $E(z)$ and $E^*(s)$, the theorems developed in Chapter 2 for the z -transform also apply to the starred transform. In addition, z -transform tables become tables of starred transforms with the substitution $z = \epsilon^{Ts}$. For this reason, a separate table for starred transforms is usually not given.

4.3 THE PULSE TRANSFER FUNCTION

In this section we develop an expression for the z-transform of the output of open-loop sampled-data systems. This expression will be required later when we form closed-loop systems by feeding back this output signal.

Consider the open-loop system of Figure 4-1a, where $G_p(s)$ is the plant transfer function. We denote the product of the plant transfer function and the zero-order hold transfer function as $G(s)$, as shown in the figure; that is,

$$G(s) = \frac{1 - e^{-Ts}}{s} G_p(s)$$

Hence this system can be represented as shown in Figure 4-1b. Note that when a representation of a system as shown in Figure 4-1b is given, $G(s)$ must contain the transfer function of a data hold. In general, *we do not show* the data-hold transfer function separately but combine it with the transfer function of that part of the system that follows the data hold.

In Figure 4-1,

$$C(s) = G(s)E^*(s) \quad (4-5)$$

Assume that $c(t)$ is continuous at all sampling instants. From (3-11),

$$C^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} C(s + jn\omega_s) = [G(s)E^*(s)]^* \quad (4-6)$$

where $[\cdot]^*$ denotes the starred transform of the function in the brackets. Thus, from (4-5) and (4-6),

$$C^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} G(s + jn\omega_s)E^*(s + jn\omega_s) \quad (4-7)$$

Equation (3-20) gives the periodic property

$$E^*(s + jn\omega_s) = E^*(s)$$

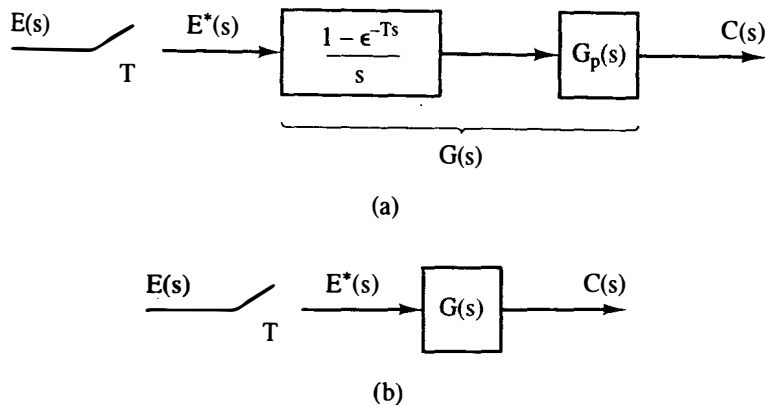


Figure 4-1 Open-loop sampled-data system.

Thus (4-7) becomes

$$C^*(s) = E^*(s) \frac{1}{T} \sum_{n=-\infty}^{\infty} G(s + jn\omega_s) = E^*(s)G^*(s) \quad (4-8)$$

and then from (4-3),

$$C(z) = E(z)G(z) \quad (4-9)$$

Now $G(z)$ is called the *pulse transfer function* and is the transfer function between the sampled input and the output *at the sampling instants*. Note that the pulse transfer function gives no information on the nature of the output, $c(t)$, between sampling instants. This information is not contained in either (4-8) or (4-9). However, we generally choose the sample frequency such that the response at sampling instants gives a very good indication of the response between sampling instants.

The derivation above is completely general. Thus given any function that can be expressed as

$$A(s) = B(s)F^*(s) \quad (4-10)$$

where $F^*(s)$ must be expressible as

$$F^*(s) = f_0 + f_1 e^{-Ts} + f_2 e^{-2Ts} + \dots$$

then, from the preceding development,

$$A^*(s) = B^*(s)F^*(s) \quad (4-11)$$

Hence, from (4-3),

$$A(z) = B(z)F(z) \quad (4-12)$$

where $B(s)$ is a function of s and $F^*(s)$ is a function of e^{Ts} ; that is, in $F^*(s)$, s appears only in the form e^{Ts} . Then, in (4-12),

$$B(z) = \mathcal{Z}[B(s)], \quad F(z) = F^*(s)|_{e^{Ts}=z} \quad (4-13)$$

The following examples illustrate this procedure.

Example 4.2

Suppose that we wish to find the z -transform of

$$A(s) = \frac{1 - e^{-Ts}}{s(s+1)} = \frac{1}{s(s+1)}(1 - e^{-Ts})$$

From (4-10), we consider

$$B(s) = \frac{1}{s(s+1)}$$

and

$$F^*(s) = 1 - e^{-Ts} \Rightarrow F(z) = 1 - z^{-1} = \frac{z-1}{z}$$

Then, from the z -transform tables,

$$B(z) = \mathcal{Z}\left[\frac{1}{s(s+1)}\right] = \frac{(1 - \epsilon^{-T})z}{(z-1)(z - \epsilon^{-T})}$$

and

$$A(z) = B(z)F(z) = \frac{(1 - \epsilon^{-T})z}{(z-1)(z - \epsilon^{-T})} \left[\frac{z-1}{z} \right] = \frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}}$$

Example 4.3

Given the system shown in Figure 4-2, with input $e(t)$ a unit step function, let us determine the output function $C(z)$. Now,

$$C(s) = \frac{1 - \epsilon^{-Ts}}{s(s+1)} E^*(s) = G(s)E^*(s)$$

In Example 4.2 it was shown that

$$G(z) = \mathcal{Z}\left[\frac{1 - \epsilon^{-Ts}}{s(s+1)}\right] = \frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}}$$

In addition, from the table in Appendix VIII,

$$E(z) = \mathcal{Z}[u(t)] = \frac{z}{z-1}$$

Thus

$$C(z) = G(z)E(z) = \frac{z(1 - \epsilon^{-T})}{(z-1)(z - \epsilon^{-T})} = \frac{z}{z-1} - \frac{z}{z - \epsilon^{-T}}$$

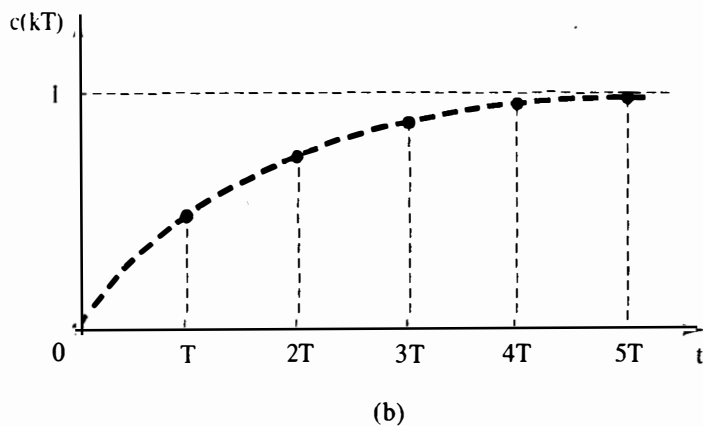
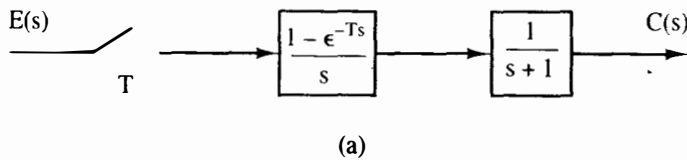


Figure 4-2 Sampled-data system.

and the inverse z -transform of this function yields

$$c(nT) = 1 - e^{-nT}$$

This response is plotted in Figure 4-2b.

Note that the output in Figure 4-2b rises exponentially to a final value of unity at the sampling instants. Note also that the z -transform analysis yields the response *only* at the sampling instants. Based on the preceding analysis, we know nothing about the response of the system between sampling instants. If this information is needed (it normally is in a practical situation), we usually find the complete response by simulation.

If the input to the sampler/zero-order hold is a unit step signal, the output is also a unit step signal. Thus the zero-order hold reconstructs the sampled step function exactly. Because of this, the response of the system of Figure 4-2 is simply the step response of a continuous-time system with a transfer function of $1/(s + 1)$. The reader can calculate this response to be

$$c(t) = 1 - e^{-t}$$

The z -transform analysis of the last example is then seen to be correct. Given $c(t)$, we can find $c(nT)$ by replacing t with nT . However, given $c(nT)$ from a z -transform analysis, we *cannot* replace nT with t and have the correct expression for $c(t)$, in general.

A final point will be made relative to the example above. For many control systems, the steady-state gain for a constant input is important; we will call this gain the *dc gain*. For a system of the configuration of Figure 4-1, the steady-state output, $c_{ss}(k)$, for an input of unity [$E(z) = z/(z - 1)$], is, from the final-value property,

$$\begin{aligned} c_{ss}(k) &= \lim_{z \rightarrow 1} (z - 1)C(z) = \lim_{z \rightarrow 1} (z - 1)G(z)E(z) \\ &= \lim_{z \rightarrow 1} (z - 1)G(z) \frac{z}{z - 1} = \lim_{z \rightarrow 1} G(z) = G(1) \end{aligned}$$

assuming the $c_{ss}(k)$ exists. Since the steady-state input is unity, we see that the dc gain is given by

$$\text{dc gain} = G(z)|_{z=1} = G(1)$$

For the example above, $G(1) = 1$, which checks the time-response calculation. Since, for a constant input, the gain of the sampler/zero-order-hold is unity, the dc gain of the system of Figure 4-1 is also given by

$$\text{dc gain} = \lim_{s \rightarrow 0} G_p(s)$$

Thus

$$\text{dc gain} = \lim_{z \rightarrow 1} G(z) = \lim_{s \rightarrow 0} G_p(s) \quad (4-14)$$

This relationship gives us a relatively easy check on the calculation of $G(z)$, since each term in (4-14) is easily evaluated. For example, in Example 4.3,

$$\lim_{z \rightarrow 1} G(z) = \lim_{z \rightarrow 1} \frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}} = 1; \quad \lim_{s \rightarrow 0} G_p(s) = \lim_{s \rightarrow 0} \frac{1}{s + 1} = 1$$

We will now investigate open-loop systems of other configurations. Consider the system of Figure 4-3a. In this system, there are two plants, and both $G_1(s)$ and $G_2(s)$ contain the transfer functions of the data holds. Now,

$$C(s) = G_2(s)A^*(s)$$

and thus

$$C(z) = G_2(z)A(z) \quad (4-15)$$

Also,

$$A(s) = G_1(s)E^*(s)$$

and thus

$$A(z) = G_1(z)E(z) \quad (4-16)$$

Then, from (4-15) and (4-16),

$$C(z) = G_1(z)G_2(z)E(z) \quad (4-17)$$

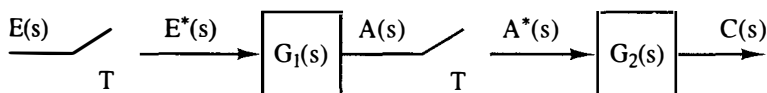
and the total transfer function is the product of the pulse transfer functions.

Consider now the system of Figure 4-3b. Of course, in this case, $G_2(s)$ would not contain a data-hold transfer function. Then

$$C(s) = G_1(s)G_2(s)E^*(s)$$

and

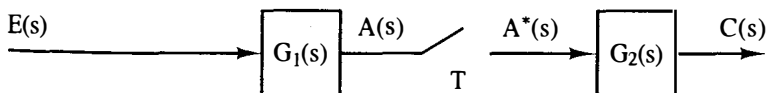
$$C(z) = \overline{G_1 G_2}(z)E(z)$$



(a)



(b)



(c)

Figure 4-3 Open-loop sampled-data systems.

where

$$\overline{G_1 G_2}(z) = \mathfrak{z}[G_1(s)G_2(s)] \quad (4-18)$$

The bar above a product term indicates that the product must be performed in the s -domain before the z -transform is taken. In addition, note that

$$\overline{G_1 G_2}(z) \neq G_1(z)G_2(z) \quad (4-19)$$

that is, the z -transform of a product of functions is not equal to the product of the z -transforms of the functions.

For the system of Figure 4-3c,

$$C(s) = G_2(s)A^*(s) = G_2(s)\overline{G_1 E^*}(s)$$

Thus

$$C(z) = G_2(z)\overline{G_1 E}(z) \quad (4-20)$$

For this system a transfer function cannot be written; that is, we cannot factor $E(z)$ from $\overline{G_1 E}(z)$. $E(z)$ contains the values of $e(t)$ only at $t = kT$. But the signal $a(t)$ in Figure 4-3c is a function of all previous values of $e(t)$, not just the values at sampling instants. Since

$$A(s) = G_1(s)E(s)$$

then, from the convolution property of the Laplace transform,

$$a(t) = \int_0^t g_1(t - \tau)e(\tau) d\tau \quad (4-21)$$

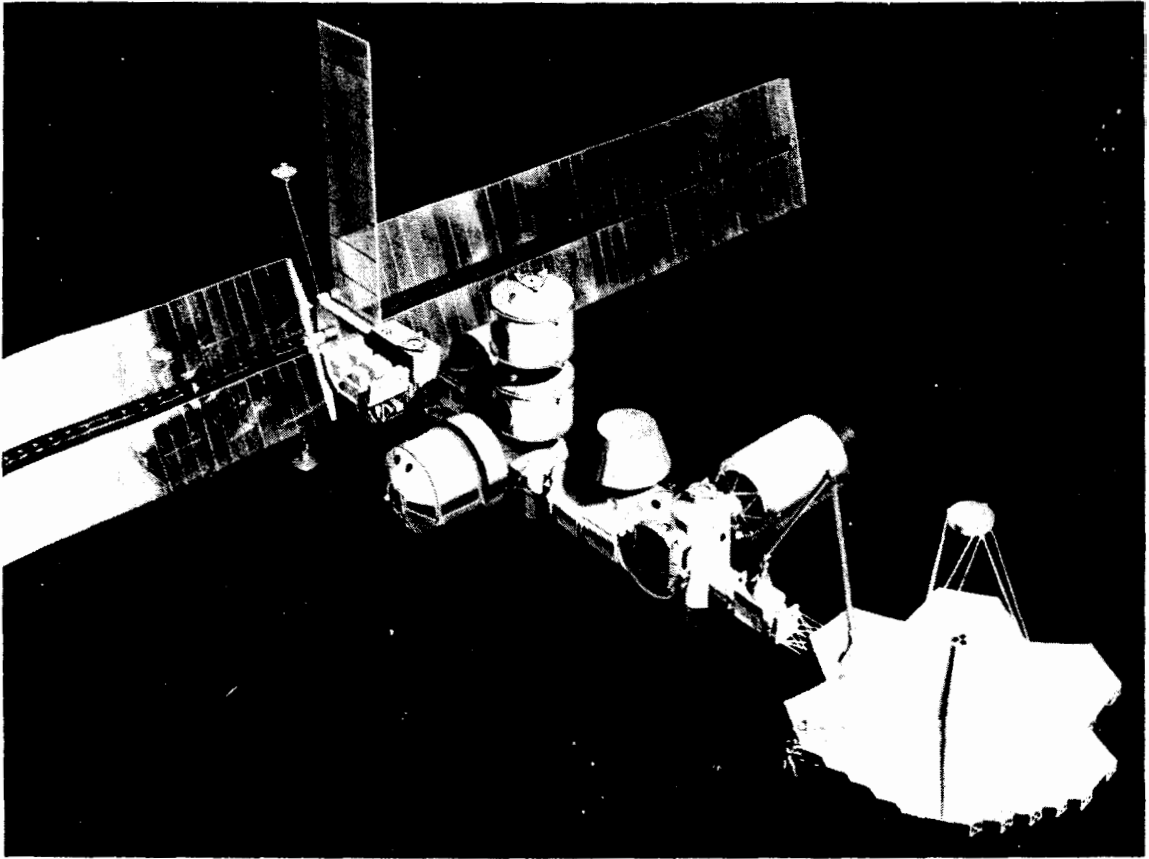
and the dependency of $a(t)$ on all previous value of $e(t)$ is seen. In general, if the input to a sample-data system is applied directly to a continuous-time part of the system before being sampled, the z -transform of the output of the system cannot be expressed as a function of the z -transform of the input signal. We will see later that this type of system presents no special problems in analysis and design.

4.4 OPEN-LOOP SYSTEMS CONTAINING DIGITAL FILTERS

In the preceding section a transfer-function technique was developed for open-loop sampled-data systems. In this section this technique is extended to cover the case in which the open-loop sampled-data system contains a digital filter.

In the system shown in Figure 4-4, the A/D converter on the filter input converts the continuous-time signal $e(t)$ to a number sequence $\{e(kT)\}$; the digital filter processes this number sequence $\{e(kT)\}$ and generates the output number sequence $\{m(kT)\}$, which in turn is converted to the continuous-time signal $\overline{m}(t)$ by the D/A converter.

As was shown in Chapter 2, a digital filter that solves a linear difference equation with constant coefficients can be represented by a transfer function $D(z)$,



A manned space platform, such as this one proposed by NASA, will utilize many digital control systems. (Courtesy of NASA.)

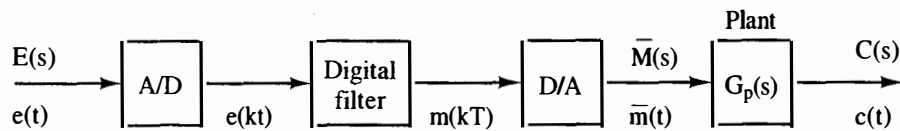


Figure 4-4 Open-loop system with a digital filter.

such that

$$M(z) = D(z)E(z) \quad (4-22)$$

or, through the substitution $z = \epsilon^{sT}$,

$$M^*(s) = D^*(s)E^*(s)$$

The D/A converter usually has an output data-hold register which gives the D/A the characteristics of a zero-order hold, and $\bar{m}(t)$ is of the form $\bar{e}(t)$ in Figure 3-3. Thus the Laplace transform of the signal $\bar{m}(t)$ can be expressed as, from (3-2),

$$\bar{M}(s) = \frac{1 - \epsilon^{-Ts}}{s} M^*(s)$$

Hence

$$C(s) = G_p(s)\bar{M}(s) = G_p(s)\frac{1 - \epsilon^{-Ts}}{s}M^*(s)$$

Then, from (4-22),

$$C(s) = G_p(s)\frac{1 - \epsilon^{-Ts}}{s}D(z)|_{z=\epsilon^{Ts}}E^*(s) \quad (4-23)$$

and we see that the filter and associated A/D and D/A converters can be represented in block-diagram form as shown in Figure 4-5. Hence, from (4-23) or Figure 4-5,

$$C(z) = \mathfrak{Z}\left[G_p(s)\frac{1 - \epsilon^{-Ts}}{s}\right]D(z)E(z) = G(z)D(z)E(z) \quad (4-24)$$

The digital computing device which implements the digital filter in Figure 4-4 actually processes the values of the input data samples $\{e(kT)\}$. However, our model for the digital filter processes a sequence of *impulse functions* of weight $\{e(kT)\}$. Hence the complete model *must be used* as depicted in Figure 4-5; that is, the combination of an ideal sampler, $D(z)$, and a zero-order hold does accurately model the combination of the A/D, digital filter, and D/A.

Example 4.4

Let us determine the step response of the system shown in Figure 4-5. Suppose that the filter is described by the difference equation

$$m(kT) = 2e(kT) - e[(k-1)T]$$

and thus

$$D(z) = \frac{M(z)}{E(z)} = 2 - z^{-1} = \frac{2z - 1}{z}$$

In addition, suppose that

$$G_p(s) = \frac{1}{s+1}$$

Then, as shown in Example 4.3,

$$\mathfrak{Z}\left[\frac{1 - \epsilon^{-Ts}}{s(s+1)}\right] = \frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}}$$

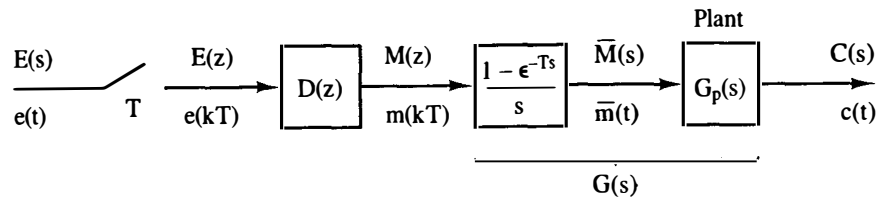


Figure 4-5 Model for the open-loop system.

Since $E(z) = z/(z - 1)$, from (4-24) we obtain

$$\begin{aligned} C(z) &= D(z)G(z)E(z) \\ &= \frac{2z - 1}{z} \left[\frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}} \right] \frac{z}{z - 1} = \frac{(2z - 1)(1 - \epsilon^{-T})}{(z - 1)(z - \epsilon^{-T})} \end{aligned}$$

By partial fractions,

$$\frac{C(z)}{z} = \frac{(2z - 1)(1 - \epsilon^{-T})}{z(z - 1)(z - \epsilon^{-T})} = \frac{1 - \epsilon^T}{z} + \frac{1}{z - 1} + \frac{\epsilon^T - 2}{z - \epsilon^{-T}}$$

and

$$C(z) = (1 - \epsilon^T) + \frac{z}{z - 1} + \frac{(\epsilon^T - 2)z}{z - \epsilon^{-T}}$$

Since

$$\mathcal{Z}^{-1}[k_i z^{-i}] = \begin{cases} k_i, & n = i \\ 0, & n \neq i \end{cases}$$

the inverse z -transform of the first term in $C(z)$ has a nonzero value only for $n = 0$, and thus

$$c(nT) = 1 + (\epsilon^T - 2)\epsilon^{-nT}, \quad n = 1, 2, 3, \dots$$

and

$$c(0) = 1 - \epsilon^T + 1 + \epsilon^T - 2 = 0$$

Note that the value $c(0) = 0$ is obvious from inspection of $C(z)$, since the order of the numerator is less than the order of the denominator. The verification of this example is considered in Problem 4-11.

We can partially verify the response of the system of Example 4.4. Note that the final value of $c(nT)$ is unity. From the final value property of the z -transform, we verify this value:

$$\lim_{n \rightarrow \infty} c(nT) = \lim_{z \rightarrow 1} (z - 1)C(z) = \lim_{z \rightarrow 1} \frac{(2z - 1)(1 - e^{-T})}{z - \epsilon^{-T}} = 1$$

In addition, from (4-14), the dc gain of the system is given by

$$\text{dc gain} = D(z) \Big|_{z=1} G_p(s) \Big|_{s=0} = \frac{2z - 1}{z} \Big|_{z=1} \frac{1}{s + 1} \Big|_{s=0} = 1$$

Since in the steady state, the input is constant at a value of unity, the steady-state value of the output is

$$\lim_{n \rightarrow \infty} c(nT) = (\text{dc gain})(\text{constant input}) = (1)(1) = 1$$

which once again verifies the steady-state output.

As a final point in this example, note that the output of the digital filter is given by

$$M(z) = (2 - z^{-1})E(z) \Rightarrow m(kT) = 2u(kT) - u[(k - 1)T]$$

Hence in the steady state, the plant input is constant at a value of unity. It was shown in Example 4.3 that this condition results in a steady-state output of unity.

4.5 THE MODIFIED z -TRANSFORM

The analysis of open-loop systems, including those that contain digital filters, has been presented in preceding sections. However, this technique of analysis does not apply to systems containing ideal time delays. To analyze systems of this type, it is necessary to define the z -transform of a delayed time function. This transform is called the modified z -transform, which is developed in this section.

The modified z -transform can be developed by considering a time function $e(t)$ that is delayed by an amount ΔT , $0 < \Delta \leq 1$, that is, by considering $e(t - \Delta T)u(t - \Delta T)$. The ordinary z -transform of the delayed time function is

$$\mathcal{Z}[e(t - \Delta T)u(t - \Delta T)] = \mathcal{Z}[E(s)\epsilon^{-\Delta Ts}] = \sum_{n=1}^{\infty} e(nT - \Delta T)z^{-n} \quad (4-25)$$

Note that the sampling is not delayed; that is, the sampling instants are $t = 0, T, 2T, \dots$. The z -transform in (4-25) is called the delayed z -transform, and thus, by definition, the delayed z -transform of $e(t)$ is

$$E(z, \Delta) = \mathcal{Z}[e(t - \Delta T)u(t - \Delta T)] = \mathcal{Z}[E(s)\epsilon^{-\Delta Ts}] \quad (4-26)$$

The delayed starred transform is also defined in (4-26), with the substitution $z = \epsilon^{Ts}$. The delayed z -transform will now be illustrated by an example.

Example 4.5

Find $E(z, \Delta)$, if $\Delta = 0.4$, for $e(t) = \epsilon^{-at}u(t)$. From (4-25),

$$\begin{aligned} E(z, \Delta) &= \epsilon^{-0.6aT}z^{-1} + \epsilon^{-1.6aT}z^{-2} + \epsilon^{-2.6aT}z^{-3} + \dots \\ &= \epsilon^{-0.6aT}z^{-1}[1 + \epsilon^{-aT}z^{-1} + \epsilon^{-2aT}z^{-2} + \dots] \\ &= \frac{\epsilon^{-0.6aT}z^{-1}}{1 - \epsilon^{-aT}z^{-1}} = \frac{\epsilon^{-0.6aT}}{z - \epsilon^{-aT}} \end{aligned}$$

The sketches in Figure 4-6 show both $e(t)$ and $e(t - \Delta T)$.

The modified z -transform is defined from the delayed z -transform. By definition, the modified z -transform of a function is equal to the delayed z -transform with Δ replaced by $1 - m$. Thus if we let $E(z, m)$ be the modified z -transform of $E(s)$, then, from (4-26),

$$E(z, m) = E(z, \Delta)|_{\Delta=1-m} = \mathcal{Z}[E(s)\epsilon^{-\Delta Ts}]|_{\Delta=1-m} \quad (4-27)$$

From (4-25) and (4-27),

$$\begin{aligned} E(z, m) &= [e(T - \Delta T)z^{-1} + e(2T - \Delta T)z^{-2} \\ &\quad + e(3T - \Delta T)z^{-3} + \dots]_{\Delta=1-m} \\ &= e(mT)z^{-1} + e[(1 + m)T]z^{-2} + e[(2 + m)T]z^{-3} + \dots \end{aligned} \quad (4-28)$$

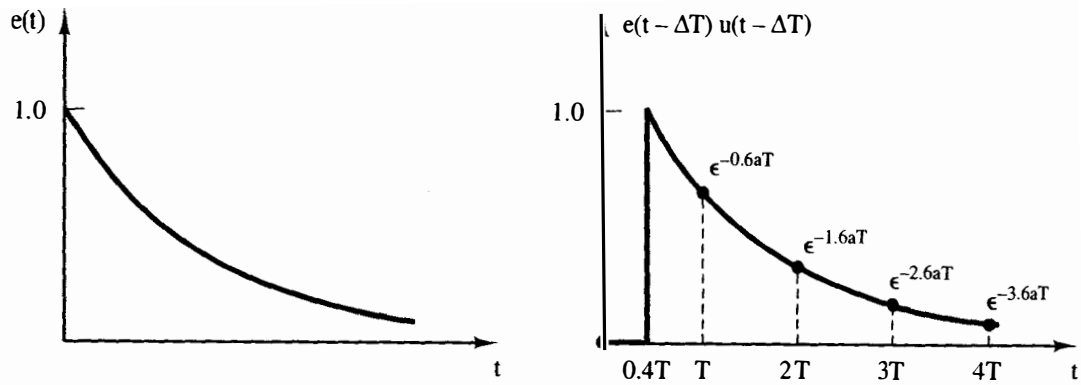


Figure 4-6 Example of the delayed z-transform.

with $\Delta = 1 - m$. Two properties of the modified z-transform are

$$E(z, 1) = E(z, m)|_{m=1} = E(z) - e(0) \quad (4-29)$$

and

$$E(z, 0) = E(z, m)|_{m=0} = z^{-1} E(z) \quad (4-30)$$

The value $m = 1$ denotes no delay [but $e(0)$ does not appear in (4-28)], and the value $m = 0$ denotes a delay of one sample period.

Example 4.6

It is desired to find the modified z-transform of $e(t) = e^{-t}$. From (4-28),

$$\begin{aligned} E(z, m) &= e^{-mT} z^{-1} + e^{-(1+m)T} z^{-2} + e^{-(2+m)T} z^{-3} + \dots \\ &= e^{-mT} z^{-1} [1 + e^{-T} z^{-1} + e^{-2T} z^{-2} + \dots] \\ &= \frac{e^{-mT} z^{-1}}{1 - e^{-T} z^{-1}} = \frac{e^{-mT}}{z - e^{-T}} \end{aligned}$$

A sketch of the time functions is given in Figure 4-7.

The ordinary z-transform tables do not apply to the modified z-transform. Instead, special tables must be derived. These tables are obtained by the following development. One can express the modified z-transform of a function $E(s)$ in the following manner, from the development in Appendix III.

$$E(z, m) = \mathcal{Z}[E(s)e^{-\Delta Ts}]|_{\Delta=1-m} = \mathcal{Z}[E(s)e^{-(1-m)Ts}] = z^{-1} \mathcal{Z}[E(s)e^{mTs}] \quad (4-31)$$

Now $\lim_{\lambda \rightarrow \infty} \lambda E(\lambda) e^{mT\lambda}$ is zero in the second integral in (A3-7) (see Figure A3-2), and thus (3-10) applies. Then, from (3-10) and (4-31),

$$E(z, m) = z^{-1} \sum_{\text{poles of } E(\lambda)} \text{residues of } E(\lambda) e^{mT\lambda} \frac{1}{1 - z^{-1} e^{T\lambda}} \quad (4-32)$$

Also, from (3-11),

$$E^*(s, m) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) e^{-(1-m)(s + jn\omega_s)T} \quad (4-33)$$

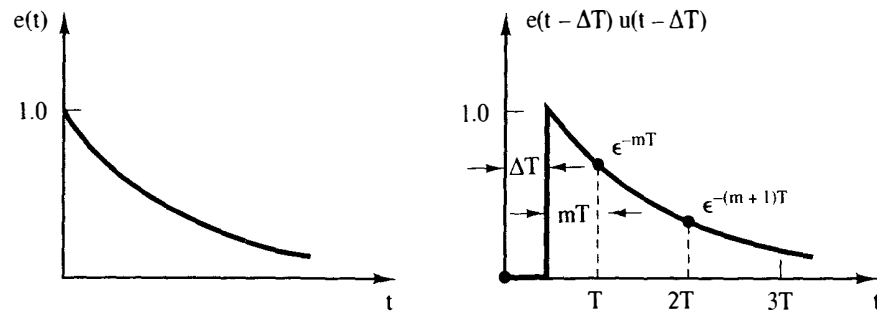


Figure 4-7 Example of the modified z-transform.

provided that $e(t - \Delta T)$ is continuous at all sampling instants. Modified z-transform tables are included with the ordinary z-transform tables in Appendix VIII.

A useful property of the modified z-transform will now be stated. Since the modified z-transform is the ordinary z-transform of a shifted function, the theorems of the ordinary z-transform derived in Chapter 2 may be applied to the modified z-transform, if care is exercised. In particular, the shifting theorem applies directly. Let $\mathcal{Z}_m[\cdot]$ indicate the modified z-transform; that is,

$$\mathcal{Z}_m[E(s)] = E(z, m) = \mathcal{Z}[\epsilon^{-\Delta Ts} E(s)]|_{\Delta = 1 - m} \quad (4-34)$$

Then, by the shifting theorem, for k a positive integer,

$$\mathcal{Z}_m[\epsilon^{-kTs} E(s)] = z^{-k} \mathcal{Z}_m[E(s)] = z^{-k} E(z, m) \quad (4-35)$$

Example 4.7

We wish to find the modified z-transform of the function $e(t) = t$. It is well known that $E(s) = 1/s^2$. This function has a pole of order 2 at $s = 0$. Therefore, the modified z-transform can be obtained from (4-32) as [see (2-34)]

$$\begin{aligned} E(z, m) &= z^{-1} \left[\frac{d}{d\lambda} \left[\frac{\epsilon^{mT\lambda}}{1 - z^{-1} \epsilon^{T\lambda}} \right]_{\lambda=0} \right] \\ &= z^{-1} \left[\frac{(1 - z^{-1} \epsilon^{T\lambda}) m T \epsilon^{mT\lambda} - \epsilon^{mT\lambda} (-T z^{-1} \epsilon^{T\lambda})}{(1 - z^{-1} \epsilon^{T\lambda})^2} \right]_{\lambda=0} \\ &= z^{-1} \left[\frac{mT(1 - z^{-1}) + Tz^{-1}}{(1 - z^{-1})^2} \right] \\ &= \frac{mT(z - 1) + T}{(z - 1)^2} \end{aligned}$$

which is verified in the table in Appendix VIII.

4.6 SYSTEMS WITH TIME DELAYS

The modified z-transform may be used to determine the pulse transfer functions of discrete-time systems containing ideal time delays. To illustrate this, consider the system of Figure 4-8, which has an ideal time delay of t_0 seconds. For this system.

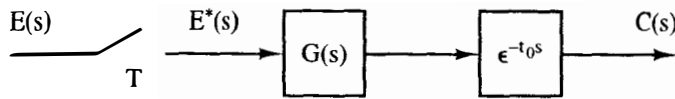


Figure 4-8 System with ideal time delay.

$$C(s) = G(s)\epsilon^{-t_0 s} E^*(s) \quad (4-36)$$

Thus

$$C(z) = \mathcal{Z}[G(s)\epsilon^{-t_0 s}]E(z) \quad (4-37)$$

If we now let

$$t_0 = kT + \Delta T, \quad 0 < \Delta < 1 \quad (4-38)$$

where k is a positive integer, then from (4-35),

$$C(z) = z^{-k} \mathcal{Z}[G(s)\epsilon^{-\Delta Ts}]E(z) = z^{-k} G(z, m)E(z) \quad (4-39)$$

where $m = 1 - \Delta$. The foregoing development will now be illustrated with an example.

Example 4.8

In Figure 4-8, let the input be a unit step, $t_0 = 0.4T$, and

$$G(s) = \frac{1 - \epsilon^{-Ts}}{s(s + 1)}$$

This system was considered in Example 4.3 with no delay. From (4-35) and the modified z -transform tables,

$$\begin{aligned} G(z, m) &= \mathcal{Z}_m \left[\frac{1 - e^{-Ts}}{s(s + 1)} \right] = (1 - z^{-1}) \mathcal{Z}_m \left[\frac{1}{s(s + 1)} \right] \\ &= \frac{z - 1}{z} \left[\frac{z(1 - \epsilon^{-mT}) + \epsilon^{-mT} - \epsilon^{-T}}{(z - 1)(z - \epsilon^{-T})} \right] \end{aligned}$$

Thus, since $mT = T - \Delta T = 0.6T$,

$$G(z, m) = \frac{z - 1}{z} \left[\frac{z(1 - \epsilon^{-0.6T}) + \epsilon^{-0.6T} - \epsilon^{-T}}{(z - 1)(z - \epsilon^{-T})} \right]$$

Since $k = 0$ in (4-39), $C(z)$ is seen to be

$$C(z) = G(z, m) \frac{z}{z - 1} = \frac{z(1 - \epsilon^{-0.6T}) + \epsilon^{-0.6T} - \epsilon^{-T}}{(z - 1)(z - \epsilon^{-T})}$$

By the power-series method of Section 2.6,

$$C(z) = (1 - \epsilon^{-0.6T})z^{-1} + (1 - \epsilon^{-1.6T})z^{-2} + (1 - \epsilon^{-2.6T})z^{-3} + \dots$$

From Example 4.3, the response of this system with no delay is $c(nT) = 1 - \epsilon^{-nT}$. This response delayed by $0.4T$ is then

$$c(nT)|_{n \leftarrow (n - 0.4)} = 1 - \epsilon^{-(n - 0.4)T}, \quad n \geq 1 \quad (4-40)$$

which checks the results of this example.

The modified z -transform may also be used to determine the pulse transfer functions of digital control systems in which the computation time of the digital computer cannot be neglected. As given in (2-4), an n th-order linear digital controller solves the difference equation

$$m(k) = b_n e(k) + b_{n-1} e(k-1) + \cdots + b_0 e(k-n) - a_{n-1} m(k-1) - \cdots - a_0 m(k-n) \quad (4-41)$$

every T seconds. Let the time required for the digital controller to compute (4-41) be t_0 seconds. Thus an input at $t = 0$ produces an output at $t = t_0$, an input at $t = T$ produces an output at $t = T + t_0$, and so on. Hence the digital controller may be modeled as a digital controller without time delay, followed by an ideal time delay of t_0 seconds, as shown in Figure 4-9a. An open-loop system containing this controller may be modeled as shown in Figure 4-9b. For this system,

$$C(z) = \mathcal{Z}[G(s)\epsilon^{-t_0 s}]D(z)E(z) \quad (4-42)$$

If we let

$$t_0 = kT + \Delta T, \quad 0 < \Delta < 1 \quad (4-43)$$

with k a positive integer, then from (4-39) and (4-42) we obtain

$$C(z) = z^{-k} G(z, m) D(z) E(z) \quad (4-44)$$

where $m = 1 - \Delta$.

Example 4.9

Consider the system of Figure 4-10. This system is that of Example 4.4, with a computational delay added for the filter. The delay is 1 ms ($t_0 = 10^{-3}$ s) and $T = 0.05$ s. Thus, for this system,

$$D(z) = \frac{2z - 1}{z}$$

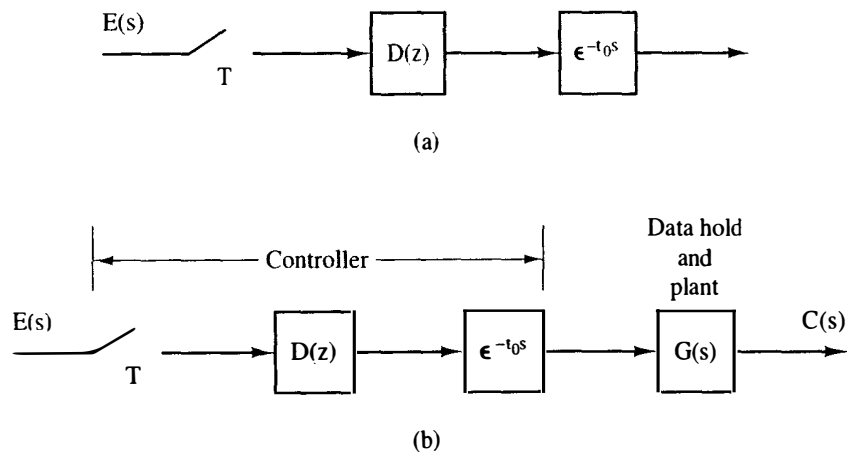


Figure 4-9 Digital controller with nonzero computation time.

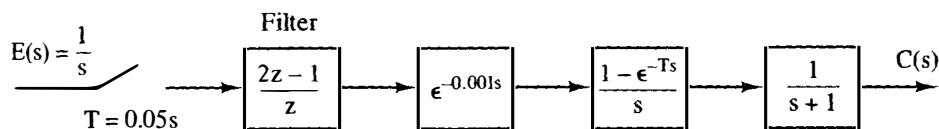


Figure 4-10 System for Example 4.9.

Now

$$mT + \Delta T = T$$

or

$$mT = T - \Delta T = 0.05 - 0.001 = 0.049$$

Then

$$G(z, m) = \mathcal{Z}_m \left[\frac{1 - e^{-Ts}}{s(s+1)} \right]_{mT=0.049} = \frac{z-1}{z} \mathcal{Z}_m \left[\frac{1}{s(s+1)} \right]_{mT=0.049}$$

From Example 4.8,

$$G(z, m) = \frac{z-1}{z} \left[\frac{z(1 - e^{-0.049}) + (e^{-0.049} - e^{-0.05})}{(z-1)(z - e^{-0.05})} \right]$$

Since the input is a unit step, then, from (4-44),

$$\begin{aligned} C(z) &= G(z, m)D(z)E(z) \\ &= \frac{z-1}{z} \left[\frac{z(1 - e^{-0.049}) + (e^{-0.049} - e^{-0.05})}{(z-1)(z - e^{-0.05})} \right] \frac{2z-1}{z} \left[\frac{z}{z-1} \right] \\ &= \frac{(2z-1)[z(1 - e^{-0.049}) + (e^{-0.049} - e^{-0.05})]}{z(z-1)(z - e^{-0.05})} \end{aligned}$$

4.7 NONSYNCHRONOUS SAMPLING

In the preceding sections, simple open-loop systems and open-loop systems with digital filters and/or ideal time delays were considered. In this section open-loop systems with nonsynchronous sampling are analyzed. *Nonsynchronous sampling* can be defined by considering the system of Figure 4-11. In this system both samplers operate at the same rate, but are not synchronous. We will now show that the output of this system can be derived using the modified z-transform.

To develop a method of analysis for systems with nonsynchronous sampling, consider the sampler and data hold in Figure 4-12a. Here the sampler operates at $hT, T + hT, 2T + hT, 3T + hT, \dots$, where $0 < h < 1$. The data-hold output is as shown in Figure 4-12b, and may be expressed as

$$\begin{aligned} \tilde{e}(t) &= e(hT)[u(t - hT) - u(t - T - hT)] + e(T + hT)[u(t - T - hT) \\ &\quad - u(t - 2T - hT)] + e(2T + hT)[u(t - 2T - hT) \\ &\quad - u(t - 3T - hT)] + \dots \end{aligned} \quad (4-45)$$

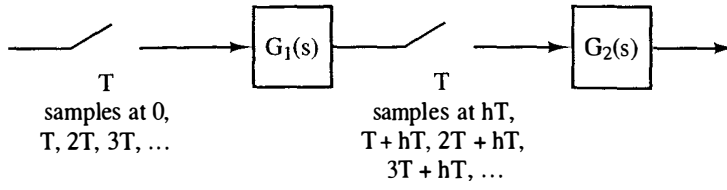


Figure 4-11 System with nonsynchronous sampling.

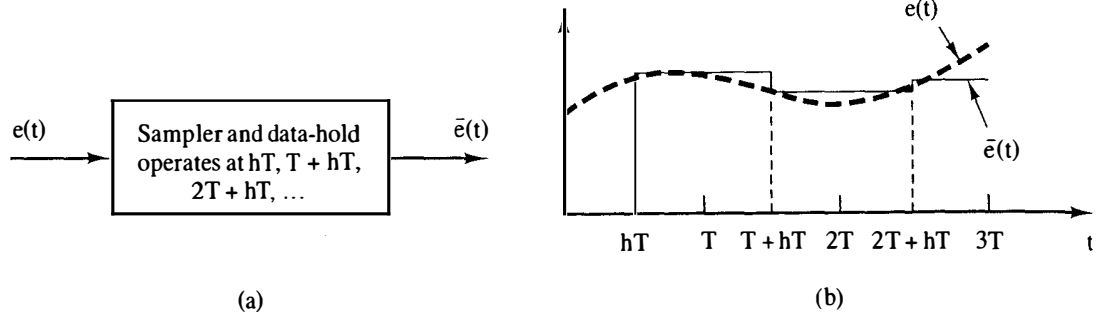


Figure 4-12 Illustration of nonsynchronous sampling.

Thus

$$\begin{aligned}\bar{E}(s) = & e(hT) \left[\frac{\epsilon^{-hTs}}{s} - \frac{\epsilon^{-(T+hT)s}}{s} \right] + e(T+hT) \left[\frac{\epsilon^{-(T+hT)s}}{s} - \frac{\epsilon^{-(2T+hT)s}}{s} \right] \\ & + e(2T+hT) \left[\frac{\epsilon^{-(2T+hT)s}}{s} - \frac{\epsilon^{-(3T+hT)s}}{s} \right] + \dots\end{aligned}$$

or

$$\begin{aligned}\bar{E}(s) &= \frac{1 - \epsilon^{-Ts}}{s} \epsilon^{-hTs} [e(hT) + e(T+hT)\epsilon^{-Ts} + e(2T+hT)\epsilon^{-2Ts} + \dots] \\ &= \frac{1 - \epsilon^{-Ts}}{s} \epsilon^{Ts} \epsilon^{-hTs} [e(hT)\epsilon^{-Ts} + e(T+hT)\epsilon^{-2Ts} + e(2T+hT)\epsilon^{-3Ts} + \dots]\end{aligned}$$

Then, from (4-28),

$$\bar{E}(s) = \frac{1 - \epsilon^{-Ts}}{s} \epsilon^{Ts} \epsilon^{-hTs} E(z, m) \Big|_{m=h, z=\epsilon^{Ts}} \quad (4-46)$$

Since

$$E(z, m) = \mathcal{Z}[E(s)\epsilon^{-\Delta Ts}] \Big|_{\Delta=1-m} \quad (4-47)$$

we see from (4-46) that the sampler and data hold of Figure 4-12 can be modeled as shown in Figure 4-13, where the sampler operates at $t = 0, T, 2T, \dots$. Note that this model delays the input signal, samples the delayed signal, and then advances the delayed sampled signal, such that the total delay in the signal is zero.

From the development above, we see that the system of Figure 4-11, with nonsynchronous samplers, may be modeled as shown in Figure 4-14. In Figure 4-14,

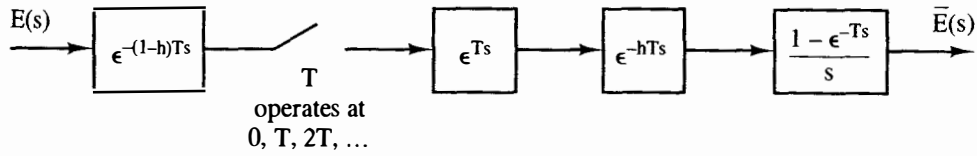


Figure 4-13 Model of a sampler and data hold.

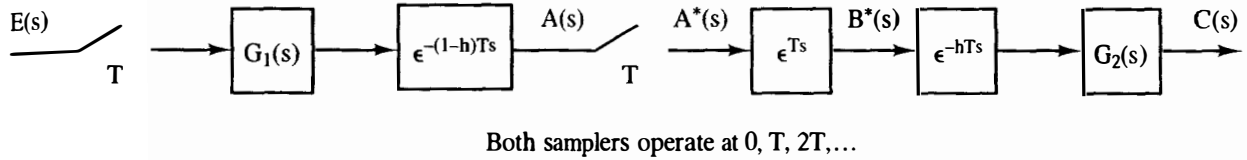


Figure 4-14 Model of the system of Figure 4-11.

the samplers are synchronous and the system is of the form of that of Figure 4-3a. Now, in Figure 4-14,

$$A(s) = e^{-(1-h)Ts} G_1(s) E^*(s)$$

and thus

$$A(z) = G_1(z, m)|_{m=h} E(z) \quad (4-48)$$

Also,

$$B^*(s) = e^{Ts} A^*(s)$$

yielding

$$B(z) = zA(z) = zE(z)G_1(z, m)|_{m=h} \quad (4-49)$$

Then

$$C(s) = e^{-hTs} G_2(s) B^*(s)$$

yielding

$$C(z) = G_2(z, m)|_{m=1-h} B(z) \quad (4-50)$$

From (4-49) and (4-50), we find $C(z)$ to be given by

$$C(z) = zE(z)G_1(z, m)|_{m=h} G_2(z, m)|_{m=1-h} \quad (4-51)$$

Example 4.10

We wish to find $C(z)$ for the system of Figure 4-15, which contains nonsynchronous sampling. Now

$$E(z) = \mathcal{Z}\left[\frac{1}{s}\right] = \frac{z}{z-1}$$

For the system, $T = 0.05$ and $hT = 0.01$. From Example 4.8,

$$G_1(z, m) = \mathcal{Z}_m\left[\frac{1 - e^{-Ts}}{s(s+1)}\right] = \frac{z-1}{z} \left[\frac{z(1 - e^{-mT}) + e^{-mT} - e^{-T}}{(z-1)(z - e^{-T})} \right]$$

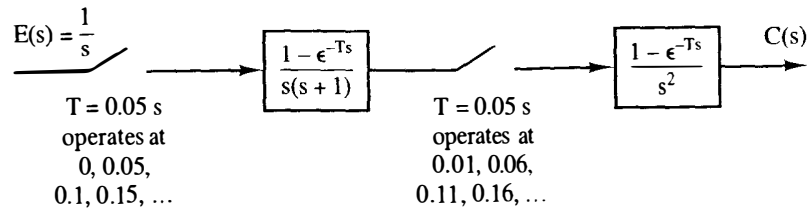


Figure 4-15 System for Example 4.10.

Then

$$G_1(z, m)|_{mT=0.01} = \frac{z-1}{z} \left[\frac{z(1 - e^{-0.01}) + e^{-0.01} - e^{-0.05}}{(z-1)(z - e^{-0.05})} \right]$$

Also, from the modified z-transform tables,

$$G_2(z, m) = \mathcal{Z}_m \left[\frac{1 - e^{-Ts}}{s^2} \right] = \frac{z-1}{z} \left[\frac{mTz - mT + T}{(z-1)^2} \right]$$

or

$$G_2(z, m)|_{m=1-h} = \frac{0.04z + 0.01}{z(z-1)}$$

Then, from the development above and (4-51),

$$\begin{aligned}
 C(z) &= z \left[\frac{z}{z-1} \right] \frac{z-1}{z} \left[\frac{z(1 - e^{-0.01}) + e^{-0.01} - e^{-0.05}}{(z-1)(z - e^{-0.05})} \right] \frac{0.04z + 0.01}{z(z-1)} \\
 &= \frac{(0.04z + 0.01)[z(1 - e^{-0.01}) + e^{-0.01} - e^{-0.05}]}{(z-1)^2(z - e^{-0.05})}
 \end{aligned}$$

4.8 STATE-VARIABLE MODELS

Thus far in this chapter we have discussed the analysis of open-loop sampled-data systems using the transfer-function approach. As was shown in Chapter 2, systems describable by a z-transform transfer function may also be modeled by discrete state equations. The state equations are of the form, from Section 2.8,

$$\begin{aligned}
 \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\
 \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k)
 \end{aligned} \tag{4-52}$$

where $\mathbf{x}(k)$ is the state vector, $\mathbf{u}(k)$ is the input vector, and $\mathbf{y}(k)$ is the output vector. The state-variable techniques used in Chapter 2 may be employed here to find the state-variable models of open-loop sampled-data systems of the type discussed in this chapter. To obtain a state-variable model:

1. Draw a simulation diagram from the z-transform transfer function.
2. Label each time-delay output as a state variable.
3. Write the state equations from the simulation diagram.

This technique will now be illustrated with an example.

Example 4.11

Consider the system of Figure 4-16a, which is the system considered in Example 4.4. Here we are denoting the output as $Y(s)$ instead of $C(s)$, to prevent any notational confusion with the C matrix. From Example 4.4,

$$G(z) = \mathcal{Z}\left[\frac{1 - e^{-Ts}}{s(s+1)}\right] = \frac{1 - e^{-T}}{z - e^{-T}}$$

and as is shown in Figure 4-16a,

$$D(z) = \frac{2z - 1}{z}$$

A simulation diagram for this system is given in Figure 4-16b. Next each delay output is labeled as a state variable. Then, from this figure we write

$$\mathbf{x}(k+1) = \begin{bmatrix} e^{-T} & -1 + e^{-T} \\ 0 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 2(1 - e^{-T}) \\ 1 \end{bmatrix} e(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

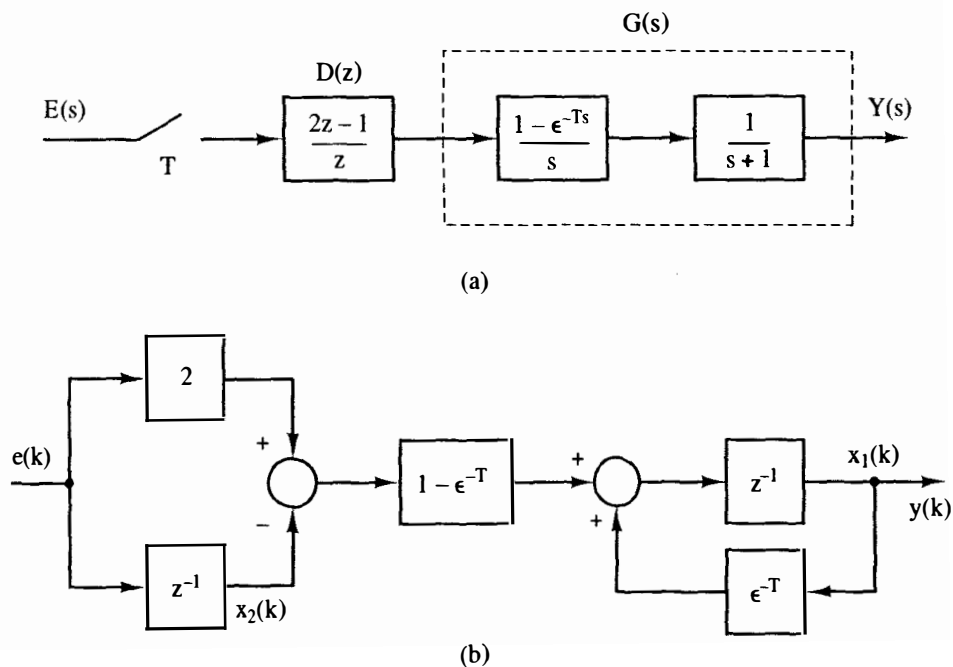


Figure 4-16 System for Example 4.11.

4.9 REVIEW OF CONTINUOUS STATE VARIABLES

Presented in the preceding section is a technique for obtaining a set of state equations describing a linear time-variant discrete system. This technique is based on transfer functions, and has two major disadvantages. One disadvantage can be illustrated by considering a simple mechanical system, the motion of a unit mass in a frictionless environment. For this system,

$$\frac{d^2 x(t)}{dt^2} = \ddot{x}(t) = f(t) \quad (4-53)$$

where $x(t)$ is the displacement, or position, of the mass, and $f(t)$ is the applied force. To obtain a continuous state-variable model, choose as state variables

$$\begin{aligned} v_1(t) &= x(t) = \text{position} \\ v_2(t) &= \dot{x}(t) = \text{velocity} \end{aligned} \quad (4-54)$$

where $\dot{x}(t) = dx(t)/dt$. We denote the states of a continuous system as $v_i(t)$, $i = 1, 2$. Then the state equations are

$$\begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f(t) \quad (4-55)$$

Here we have chosen as state variables the position and the velocity of the mass. These variables may be considered to be the "natural" states (physical variables) of the system, and would be the desirable states to choose. If this simple system were a part of a sampled-data system, we can easily choose position as one of the states in the discrete state model, by letting position be the output of the simple system. However, in taking the transfer-function approach to discrete state modeling, we would have difficulty in choosing velocity as the second state variable. Thus we lose the natural, and desirable, states of the system. Another disadvantage of the transfer-function approach is the difficulty in deriving the pulse transfer functions for high-order systems.

A different approach for obtaining the discrete state model of a system is presented in the following section. This approach is based on the use of continuous state variables. Hence a brief presentation of the requisite theory of continuous state variables will be made here.

As indicated in the example above of the motion of a mass, continuous state-variable equations for a linear-time-invariant system are of the form

$$\begin{aligned} \dot{\mathbf{v}}(t) &= \mathbf{A}_c \mathbf{v}(t) + \mathbf{B}_c \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}_c \mathbf{v}(t) + \mathbf{D}_c \mathbf{u}(t) \end{aligned} \quad (4-56)$$

In this equation, $\mathbf{v}(t)$ are the states, $\mathbf{u}(t)$ are the inputs, $\mathbf{y}(t)$ are the outputs, and the matrices are subscripted to indicate continuous state equations. Nonsubscripted matrices will indicate discrete state equations. To illustrate continuous state equations, consider the following example.

Example 4.12

It is desired to find a state model for the mechanical system of Figure 4-17a. Here M is mass, B is the damping factor for linear friction, and K is the stiffness factor for a linear spring. The equations for this system are [1]

$$\begin{aligned}\ddot{y}_1(t) + B_1 \dot{y}_1(t) + K y_1(t) + B_2 [\dot{y}_1(t) - \dot{y}_2(t)] &= 0 \\ \ddot{y}_2(t) + B_2 [\dot{y}_2(t) - \dot{y}_1(t)] &= u(t)\end{aligned}$$

The state variables are chosen as

$$\begin{aligned}v_1(t) &= y_1(t); & v_3(t) &= y_2(t) \\ v_2(t) &= \dot{y}_1(t); & v_4(t) &= \dot{y}_2(t)\end{aligned}$$

A flow graph for these equations is shown in Figure 4-17b. The transfer functions s^{-1} represent integrators, and the output of each integrator is chosen as a state, as shown. Next the state equations are written from the flow graph.

$$\dot{\mathbf{v}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K & -(B_1 + B_2) & 0 & B_2 \\ 0 & 0 & 0 & 1 \\ 0 & B_2 & 0 & -B_2 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{v}$$

In Example 4.12, the state variables are positions and velocities. If the technique of drawing a flow graph from the system differential equations (written using physical laws) is used, the states chosen will be the natural states of the system.

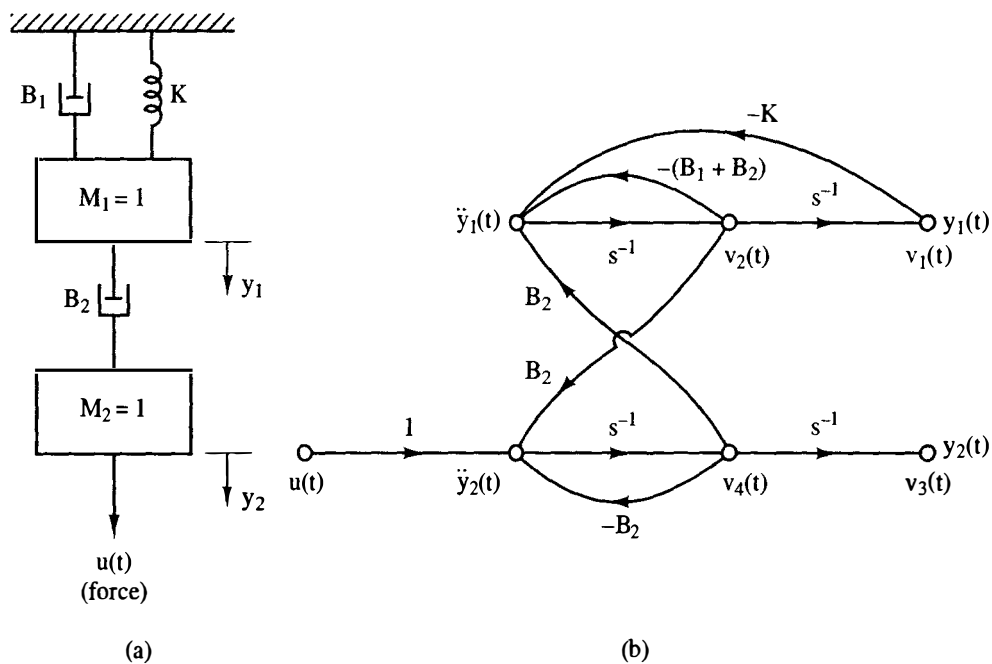


Figure 4-17 System for Example 4.12.

Consider now the state equations for a single-input, single-output continuous system.

$$\begin{aligned}\dot{\mathbf{v}}(t) &= \mathbf{A}_c \mathbf{v}(t) + \mathbf{B}_c u(t) \\ y(t) &= \mathbf{C}_c \mathbf{v}(t) + D_c u(t)\end{aligned}\quad (4-57)$$

To obtain the solution of these equations, we will use the Laplace transform. For (4-57) [1],

$$s\mathbf{V}(s) - \mathbf{v}(0) = \mathbf{A}_c \mathbf{V}(s) + \mathbf{B}_c U(s) \quad (4-58)$$

Solving for $\mathbf{V}(s)$ yields

$$\mathbf{V}(s) = [\mathbf{I}s - \mathbf{A}_c]^{-1} \mathbf{v}(0) + [\mathbf{I}s - \mathbf{A}_c]^{-1} \mathbf{B}_c U(s) \quad (4-59)$$

Define $\Phi_c(t)$ as

$$\Phi_c(t) = \mathcal{L}^{-1}\{[\mathbf{I}s - \mathbf{A}_c]^{-1}\} \quad (4-60)$$

The matrix $\Phi_c(t)$ is called the state transition matrix for (4-57). The inverse Laplace transform of (4-59) is then

$$\mathbf{v}(t) = \Phi_c(t) \mathbf{v}(0) + \int_0^t \Phi_c(t - \tau) \mathbf{B}_c u(\tau) d\tau \quad (4-61)$$

The state transition matrix $\Phi_c(t)$ can be calculated as given in (4-60). However, a different expression for $\Phi_c(t)$ may be derived. This expression is found by assuming $\Phi_c(t)$ in (4-61) to be an infinite series.

$$\Phi_c(t) = \mathbf{K}_0 + \mathbf{K}_1 t + \mathbf{K}_2 t^2 + \mathbf{K}_3 t^3 + \dots \quad (4-62)$$

where the \mathbf{K}_i are constant matrices. Choosing $u(t) = 0$, the substitution of (4-61), with $\Phi_c(t)$ given by (4-62), into (4-57) yields [1]

$$\Phi_c(t) = \mathbf{I} + \mathbf{A}_c t + \mathbf{A}_c^2 \frac{t^2}{2!} + \mathbf{A}_c^3 \frac{t^3}{3!} + \dots = \sum_{k=0}^{\infty} \frac{\mathbf{A}_c^k t^k}{k!} \quad (4-63)$$

This expression for $\Phi_c(t)$ will prove to be useful in deriving discrete state models for sampled-data systems.

A problem that often arises in determining state models of physical systems will now be discussed. This problem is illustrated by the system shown in Figure 4-18a. State equations, written from the simulation diagram of Figure 4-18b, are

$$\begin{aligned}\dot{x}(t) &= -x(t) + [u(t) - \dot{x}(t)] \\ y(t) &= \dot{x}(t)\end{aligned}$$

These equations are not in the standard format for state equations, since $\dot{x}(t)$ appears on the right-hand side of both the $\dot{x}(t)$ equation and the $y(t)$ equation. The $\dot{x}(t)$ equation must be solved for $\dot{x}(t)$, resulting in

$$\dot{x}(t) = -0.5x(t) + 0.5u(t)$$

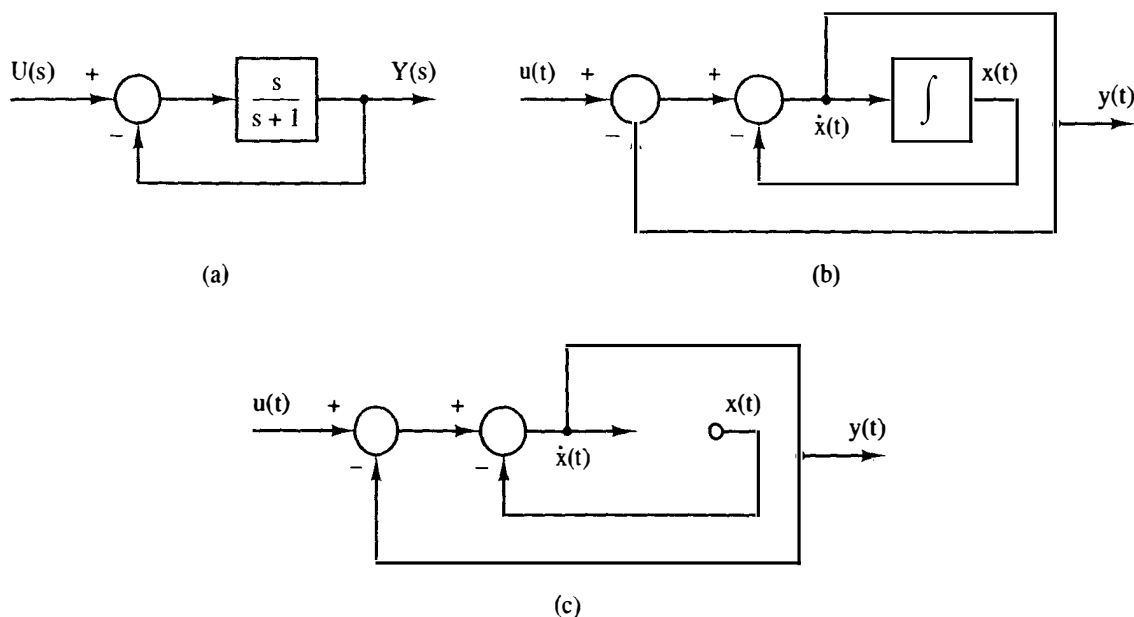


Figure 4-18 System with an algebraic loop.

and hence

$$y(t) = -0.5x(t) + 0.5u(t)$$

We now present a procedure for writing the state equations in the standard format for systems of the type shown in Figure 4-18. If we redraw the simulation diagram of Figure 4-18b with the integrator omitted, Figure 4-18c results. Now we write the equations for $\dot{x}(t)$ and $y(t)$ in terms of the inputs in this diagram, $x(t)$ and $u(t)$. This technique is standard for writing state equations from simulation diagrams, although we usually do not redraw the diagram with the integrators removed. Applying Mason's gain formula to the diagram in Figure 4-18c, we obtain the state equations

$$\dot{x}(t) = \frac{-1}{1+1}x(t) + \frac{1}{1+1}u(t) = -0.5x(t) + 0.5u(t)$$

$$y(t) = \frac{-1}{1+1}x(t) + \frac{1}{1+1}u(t) = -0.5x(t) + 0.5u(t)$$

which is the desired result.

The loop in Figure 4-18c that does not contain an integrator is called an *algebraic loop*. We now present a second procedure for writing the state equations for systems with algebraic loops. The system equations are initially written as [2]

$$\dot{\mathbf{x}}(t) = \mathbf{A}_1 \dot{\mathbf{x}}(t) + \mathbf{A}_2 \mathbf{x}(t) + \mathbf{B}_1 \mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}_1 \dot{\mathbf{x}}(t) + \mathbf{C}_2 \mathbf{x}(t) + \mathbf{D}_1 \mathbf{u}(t)$$

Solving the first equation, we obtain

$$\dot{\mathbf{x}}(t) = [\mathbf{I} - \mathbf{A}_1]^{-1} \mathbf{A}_2 \mathbf{x}(t) + [\mathbf{I} - \mathbf{A}_1]^{-1} \mathbf{B}_1 \mathbf{u}(t)$$

Then we substitute this equation into the one for $y(t)$:

$$y(t) = [C_1[I - A_1]^{-1}A_2 + C_2]x(t) + [C_1[I - A_1]^{-1}B_1 + D_1]u(t)$$

Thus we have the state equations for the system in standard form.

4.10 DISCRETE STATE EQUATIONS

A technique is developed in this section for determining the discrete state equations of a sampled-data system directly from the continuous state equations. In fact, the states of the continuous model become the states of the discrete model. Thus the natural states of the system are preserved.

To develop this technique, consider the state equations for the continuous portion of the system shown in Figure 4-19.

$$\begin{aligned}\dot{\mathbf{v}}(t) &= \mathbf{A}_c \mathbf{v}(t) + \mathbf{B}_c u(t) \\ y(t) &= \mathbf{C}_c \mathbf{v}(t) + D_c u(t)\end{aligned}\quad (4-64)$$

As shown in Section 4.9, the solution to these equations is

$$\mathbf{v}(t) = \Phi_c(t - t_0)\mathbf{v}(t_0) + \int_{t_0}^t \Phi_c(t - \tau)\mathbf{B}_c u(\tau) d\tau \quad (4-65)$$

where the initial time is t_0 , and where, from (4-63),

$$\Phi_c(t - t_0) = \sum_{k=0}^{\infty} \frac{\mathbf{A}_c^k (t - t_0)^k}{k!} \quad (4-66)$$

To obtain the discrete model we evaluate (4-65) at $t = kT + T$ with $t_0 = kT$, that is,

$$\mathbf{v}(kT + T) = \Phi_c(T)\mathbf{v}(kT) + m(kT) \int_{kT}^{kT+T} \Phi_c(kT + T - \tau)\mathbf{B}_c d\tau \quad (4-67)$$

Note that we have replaced $u(t)$ with $m(kT)$ since, during the time interval $kT \leq t < kT + T$, $u(t) = m(kT)$. It is emphasized that this development is valid only if $u(t)$ is the output of a zero-order hold.

Compare (4-67) with the discrete state equations [see (4-52)]

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}m(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) + Dm(k)\end{aligned}\quad (4-68)$$

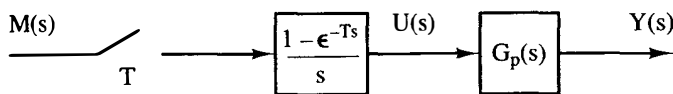


Figure 4-19 Sampled-data system.

Thus, if we let

$$\begin{aligned} \mathbf{x}(kT) &= \mathbf{v}(kT) \\ \mathbf{A} &= \Phi_c(T) \\ \mathbf{B} &= \int_{kT}^{kT+T} \Phi_c(kT + T - \tau) \mathbf{B}_c d\tau \end{aligned} \quad (4-69)$$

we obtain the discrete state equations for the sampled-data system. Hence the discrete-system \mathbf{A} and \mathbf{B} matrices are given by (4-69).

The output equation in (4-64), when evaluated at $t = kT$, yields

$$\begin{aligned} y(kT) &= \mathbf{C}_c \mathbf{v}(kT) + D_c u(kT) \\ &= \mathbf{C}_c \mathbf{x}(kT) + D_c m(kT) \end{aligned} \quad (4-70)$$

Thus the discrete \mathbf{C} and D values are equal to the continuous \mathbf{C}_c and D_c values, respectively.

The relationship for \mathbf{B} can be simplified. In (4-69) in the equation for \mathbf{B} , let $kT - \tau = -\sigma$. Then this equation becomes

$$\mathbf{B} = \left[\int_0^T \Phi_c(T - \sigma) d\sigma \right] \mathbf{B}_c \quad (4-71)$$

The discrete system matrices \mathbf{A} and \mathbf{B} may be evaluated by finding $\Phi_c(t)$ using the Laplace transform approach of (4-60). However, in general this approach is cumbersome. A more tractable technique is to use a computer evaluation of $\Phi_c(T)$ in (4-63); that is, with $t = T$, (4-66) and (4-63) becomes

$$\Phi_c(T) = \mathbf{I} + \mathbf{A}_c T + \mathbf{A}_c^2 \frac{T^2}{2!} + \mathbf{A}_c^3 \frac{T^3}{3!} + \cdots \quad (4-72)$$

Since this is a convergent series, the series can usually be truncated with adequate resulting accuracy.

The integral of (4-71), necessary for the computation of \mathbf{B} , can also be evaluated using a series expansion. In (4-71), let $\tau = T - \sigma$. Hence

$$\begin{aligned} \int_0^T \Phi_c(T - \sigma) d\sigma &= \int_T^0 \Phi_c(\tau) (-d\tau) = \int_0^T \Phi_c(\tau) d\tau \\ &= \int_0^T \left(\mathbf{I} + \mathbf{A}_c \tau + \mathbf{A}_c^2 \frac{\tau^2}{2!} + \mathbf{A}_c^3 \frac{\tau^3}{3!} + \cdots \right) d\tau \\ &= \mathbf{I}T + \mathbf{A}_c \frac{T^2}{2!} + \mathbf{A}_c^2 \frac{T^3}{3!} + \mathbf{A}_c^3 \frac{T^4}{4!} + \cdots \end{aligned} \quad (4-73)$$

and, from (4-71) and (4-73),

$$\mathbf{B} = \left[\mathbf{I}T + \mathbf{A}_c \frac{T^2}{2!} + \mathbf{A}_c^2 \frac{T^3}{3!} + \cdots \right] \mathbf{B}_c \quad (4-74)$$

Comparing (4-72) with (4-73), we see that the computer program used to evaluate $\Phi_c(T)$ may also be used to evaluate $\int_0^T \Phi_c(\tau) d\tau$, by expanding the program somewhat.

In the derivations above, only the single-input, single-output case was considered. For the multiple-input, multiple-output case, the results are the same, with D_c and D now matrices. For the multiple-input case, *each* input must be the output of a zero-order hold. In certain practical cases in which some of the inputs are not outputs of zero-order holds, the procedure above is still used. The resulting discrete model is reasonably accurate, provided that these inputs change slowly over any sample period.

The derivations above will now be illustrated by an example.

Example 4.13

For the sampled-data system of Figure 4-19, let $T = 0.1$ s and

$$G_p(s) = \frac{10}{s(s+1)}$$

For example, this plant could be a servomotor of the type described in Chapter 1. A continuous state-variable model of this system is from Figure 4-20a,

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 10 \end{bmatrix} u(t)$$

$$y(t) = [1 \ 0] \mathbf{x}(t)$$

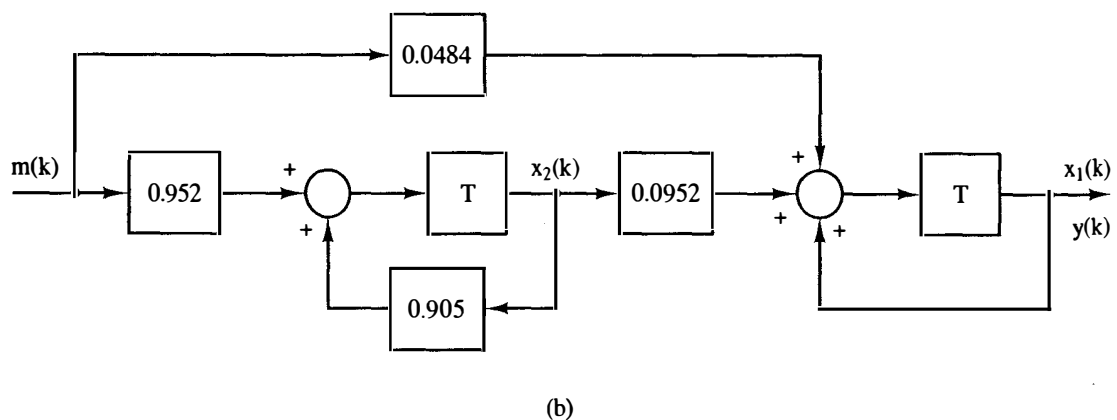
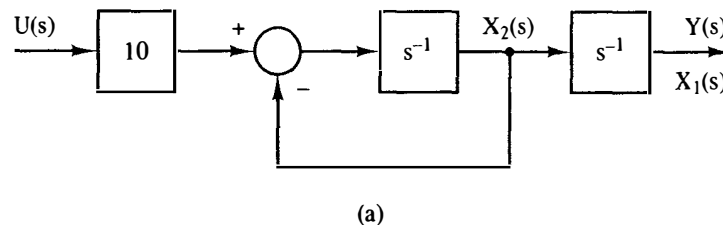


Figure 4-20 Simulation diagram for the system of Example 4.13.

For this example, since the system is second order, $\Phi_c(t)$ will be found.

$$\begin{aligned}\Phi_c(t) &= \mathcal{L}^{-1}\{[s\mathbf{I} - \mathbf{A}_c]^{-1}\} \\ &= \mathcal{L}^{-1}\begin{bmatrix} s & -1 \\ 0 & s+1 \end{bmatrix}^{-1} = \mathcal{L}^{-1}\begin{bmatrix} \frac{1}{s} & \frac{1}{s(s+1)} \\ 0 & \frac{1}{s+1} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 - e^{-t} \\ 0 & e^{-t} \end{bmatrix}\end{aligned}$$

Also,

$$\int_0^T \Phi_c(\tau) d\tau = \begin{bmatrix} \tau & \tau + e^{-\tau} \\ 0 & -e^{-\tau} \end{bmatrix}_0^T = \begin{bmatrix} T & T - 1 + e^{-T} \\ 0 & 1 - e^{-T} \end{bmatrix}$$

Then

$$\mathbf{A} = \Phi_c(T)|_{T=0.1} = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix}$$

and

$$\begin{aligned}\mathbf{B} &= \left[\int_0^T \Phi_c(\tau) d\tau \right] \mathbf{B}_c = \begin{bmatrix} 0.1 & 0.00484 \\ 0 & 0.0952 \end{bmatrix} \begin{bmatrix} 0 \\ 10 \end{bmatrix} \\ &= \begin{bmatrix} 0.0484 \\ 0.952 \end{bmatrix}\end{aligned}$$

Hence the discrete state equations are

$$\begin{aligned}\mathbf{x}(k+1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0484 \\ 0.952 \end{bmatrix} m(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k)\end{aligned}$$

A simulation diagram of this model is shown in Figure 4-20b. Figure 4-20a is the simulation diagram of the analog plant. Even though the states, the input, and the output of the two diagrams in Figure 4-20 are equal at the sampling instants, the two diagrams bear no resemblance to each other. In general, the two simulation diagrams for such a system are not similar.

The discrete matrices in the example above may also be calculated by computer. For three-significant-figure accuracy, three terms in the series expansion of $\Phi_c(t)$ [see (4-72)] are required. Five terms in the series expansion yield six-significant-figure accuracy.

4.11 PRACTICAL CALCULATIONS

As stated in the preceding section, all calculations required to develop the discrete model of an analog plant may be performed by computer. Calculation by computer is required for high-order systems, and is preferred for low-order systems to reduce errors.

We will now list the steps necessary for the computer calculations.

1. Derive the state model for the analog part of the system, in the form

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}_c \mathbf{x}(t) + \mathbf{B}_c u(t) \\ y(t) &= \mathbf{C}_c \mathbf{x}(t) + D_c u(t)\end{aligned}\quad (4-75)$$

2. If the transfer function of the analog part of the system is required, calculate (by computer) [1,3]

$$G_p(s) = \mathbf{C}_c [s\mathbf{I} - \mathbf{A}_c]^{-1} \mathbf{B}_c + D_c \quad (4-76)$$

3. Calculate the discrete matrices of the analog part of the system (by computer).

$$\begin{aligned}\mathbf{A} &= \mathbf{I} + \mathbf{A}_c T + \mathbf{A}_c^2 T^2/2! + \dots \\ [\text{eq. (4-74)}] \quad \mathbf{B} &= (\mathbf{I}T + \mathbf{A}_c T^2/2! + \mathbf{A}_c^2 T^3/3! + \dots) \mathbf{B}_c \\ \mathbf{C} &= \mathbf{C}_c, \quad D = D_c\end{aligned}\quad (4-77)$$

4. By computer, calculate the pulse transfer function [see (2-84)] using [3]

$$G(z) = \mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1} \mathbf{B} + D \quad (4-78)$$

The calculations required in steps 2, 3, and 4 are implemented in MATLAB by the statements

2. `[nc,dc] = ss2tf(Ac,Bc,Cc,Dc)`
3. `[A,B] = c2d(Ac,Bc,T)`
4. `[n,d] = ss2tf(A,B,C,D)`

In these statements, nc are the numerator-polynomial coefficients of $G_p(s)$, and dc are those of the denominator. Defined in a like manner are n and d for $G(z)$.

Note that these computer procedures give both the analog transfer function, the discrete state model, and the pulse transfer function. We do not directly use the z-transform tables in any of the steps; all steps are performed on the computer. However, we must, by some procedure, derive the analog state model. An example is now given to illustrate these computer procedures.

Example 4.14



Consider again the system of Example 4.13. A MATLAB program that calculates the discrete state matrices A, B, C, and D, and the plant transfer function $G(z) = n/d$ is given by

```
Ac = [0  1; 0 -1];
Bc = [0; 10];
C = [1  0];
D = 0;
```

```

T = 0.1;
[A,B] = c2d(Ac,Bc,T)
[n,d] = ss2tf(A,B,C,D)

```

```

result: n: 0  0.0484  0.0468    d: 1  -1.9048  0.9048

```

Only the numerator and denominator polynomial coefficients of $G(z)$ are shown displayed. Execution of the program also displays the matrices A and B .

For certain systems, the direct programming of (4-74) and (4-77) yields unacceptable numerical errors. For these cases the discrete A and B matrices must be evaluated using different algorithms [4-6].

4.12 SUMMARY

In this chapter we have examined various aspects of open-loop discrete-time systems. In particular we discussed the starred transform and showed that it possesses the properties of the z -transform defined in Chapter 2. Next, the starred transform was used to find the pulse transfer function of open-loop systems. The pulse transfer function was extended to the analysis of open-loop systems containing digital filters. In order to analyze systems containing ideal time delays, the modified z -transform and its properties were derived. Then techniques for finding discrete state-variable models of open-loop sampled-data systems were presented. These developments form the basis for the computer calculations of discrete state models and the pulse transfer function. The foundation built in this chapter for open-loop systems will serve as a basis for presenting the analysis of closed-loop discrete-time systems in Chapter 5.

REFERENCES AND FURTHER READING

1. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2d ed. Englewood Cliffs, NJ: Prentice Hall, 1991.
2. D. M. Look, "Direct State Space Formulation of Second-Order Coupled Systems," M.S. thesis, Auburn University, Auburn, AL, 1971.
3. J. L. Melsa, *Computer Programs for Computational Assistance*. New York: McGraw-Hill Book Company, 1970.
4. R. C. Ward, "Numerical Computation of the Matrix Exponential with Accuracy Estimate," *SIAM J. Numer. Anal.*, pp. 600-610, 1977.
5. C. Moler and C. Van Loam, "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Rev.*, pp. 801-836, Oct. 1978.
6. D. Westreich, "A Practical Method for Computing the Exponential of a Matrix and Its Integral," *Commun. Appl. Numer. Methods*, pp. 375-380, 1990.

7. J. A. Cadzow and H. R. Martens, *Digital-Time and Computer Control Systems*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1970.
8. P. M. DeRusso, R. J. Roy, and C. M. Close, *State Variables for Engineers*. New York: John Wiley & Sons, Inc., 1965.
9. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1988.
10. E. I. Jury, *Theory and Application of the z-Transform Method*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.
11. B. C. Kuo, *Digital Control Systems*, 2d ed. New York: Saunders College Publishing, 1992.
12. K. Ogata, *State Space Analysis of Control Systems*. Englewood Cliffs, NJ: Prentice Hall, 1967.

PROBLEMS

- 4-1. (a) Show that a pole of $E(s)$ in the left half-plane transforms into a pole of $E(z)$ inside the unit circle.
 (b) Show that a pole of $E(s)$ on the imaginary axis transforms into a pole of $E(z)$ on the unit circle.
 (c) Show that a pole of $E(s)$ in the right half-plane transforms into a pole of $E(z)$ outside the unit circle.

- 4-2. Let $T = 0.05$ s and

$$E(s) = \frac{s + 2}{(s - 1)(s + 1)}$$

- (a) Without calculating $E(z)$, find its poles.
 (b) Give the rule that you used in part (a).
 (c) Verify the results of part (a) by calculating $E(z)$.
 (d) Compare the zero of $E(z)$ with that of $E(s)$.
 (e) The poles of $E(z)$ are determined by those of $E(s)$. Does an equivalent rule exist for zeros?
- 4-3. Find the z-transform of the following functions, using z-transform tables. Compare the pole-zero locations of $E(z)$ in the z-plane with those of $E(s)$ and $E^*(s)$ in the s-plane (see Problems 3-4). Let $T = 0.1$ s.

(a) $E(s) = \frac{20}{(s + 2)(s + 5)}$

(b) $E(s) = \frac{5}{s(s + 1)}$

(c) $E(s) = \frac{s + 2}{s(s + 1)}$

(d) $E(s) = \frac{s + 2}{s^2(s + 1)}$

(e) $E(s) = \frac{s^2 + 5s + 6}{s(s + 4)(s + 5)}$

(f) $E(s) = \frac{2}{s^2 + 2s + 5}$

- (g) Verify any partial-fraction expansions required, using MATLAB.

- 4-4. Find the z-transforms of the following functions:

(a) $E(s) = \frac{(\epsilon^s - 1)^2}{\epsilon^{2s}s(s + 1)}, \quad T = 0.5$ s

$$(b) E(s) = \frac{(0.5s + 1)(1 - e^{-0.25s})}{0.5s(s + 0.25)}, \quad T = 0.25 \text{ s}$$

- 4-5. (a) Find the system response at the sampling instants to a unit step input for the system of Figure P4-5. Plot $c(nT)$ versus time.
- (b) Verify your results of (a) by determining the input to the plant, $m(t)$, and then calculating $c(t)$ by continuous-time techniques.
- (c) Find the steady-state gain for a constant input (dc gain), from both the pulse transfer function and from the plant transfer function.
- (d) Is the gain in part (c) obvious from the results of parts (a) and (b)? Why?

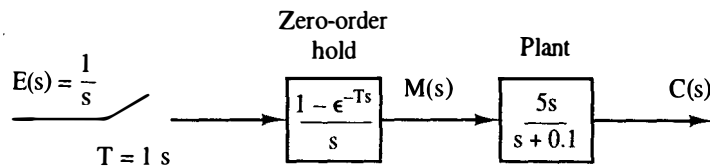


Figure P4-5 System for Problem 4-5.

- 4-6. Repeat Problem 4-5 for the case that $T = 0.1 \text{ s}$ and the plant transfer function is given by:

$$(a) G_p(s) = \frac{5}{s^2 + 3s + 2}$$

$$(b) G_p(s) = \frac{5}{s^2 + 2s + 2}$$

- (c) Verify each $G(z)$ by computer.
- 4-7. (a) Find the conditions on a transfer function $G(z)$ such that its dc gain is zero. Prove your result.
- (b) For

$$G(z) = z \left[\frac{1 - e^{-Ts}}{s} G_p(s) \right]$$

find the conditions of $G_p(s)$ such that the dc gain of $G(z)$ is zero. Prove your result.

- (c) Normally, a pole at the origin in the s -plane transforms into a pole at $z = 1$ in the z -plane. The function in the brackets in part (b) has a pole at the origin in the s -plane. Why does this pole *not* transform into a pole at $z = 1$?
- (d) Find the conditions on $G(z)$ such that its dc gain is unbounded. Prove your result.
- (e) For $G(z)$ as given in part (b), find the conditions of $G_p(s)$ such that the dc gain of $G(z)$ is unbounded. Prove your result.
- 4-8. Find the system response at the sampling instants to a unit-step input for the system of Figure P4-8.

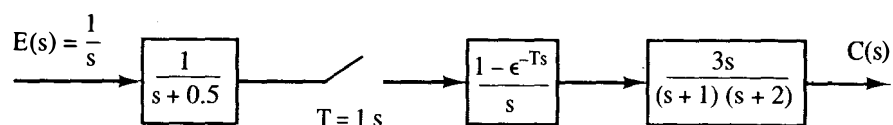


Figure P4-8 System for Problem 4-8.

- 4-9. (a) Find the output $c(kT)$ for the system of Figure P4-9, for $e(t)$ equal to a unit-step function.
 (b) What is the effect on $c(kT)$ of the sampler and data hold in the upper path? Why?
 (c) Sketch the unit-step response $c(t)$ of the system of Figure P4-9. This sketch can be made without mathematically solving for $C(s)$.
 (d) Repeat part (c) for the case that the sampler and data hold in the upper path is removed.

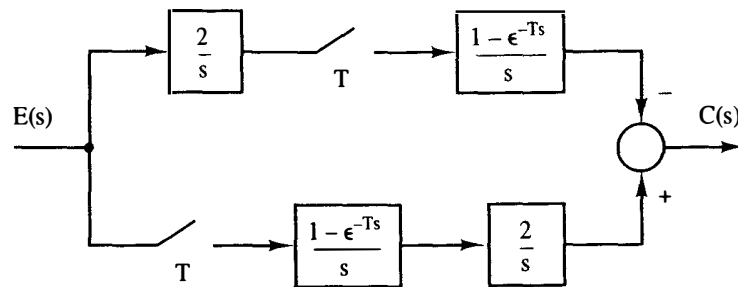
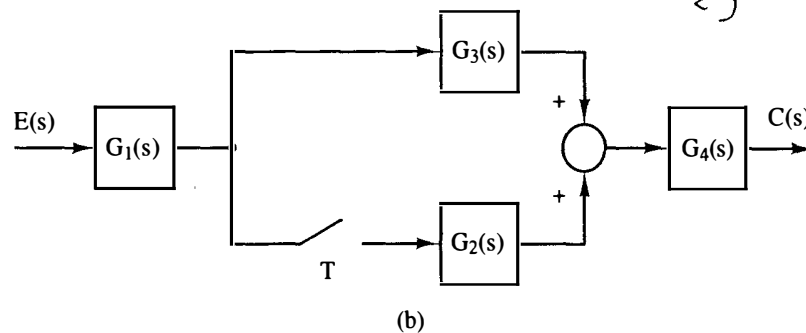
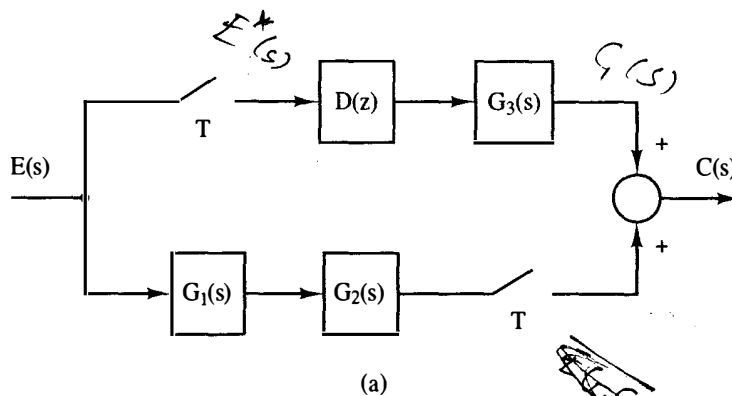


Figure P4-9 System for Problem 4-9.

- 4-10. (a) Express each $C(s)$ and $C(z)$ as functions of the input for the systems of Figure P4-10.
 (b) List those transfer functions in Figure P4-10 that contain the transfer function of a data hold.



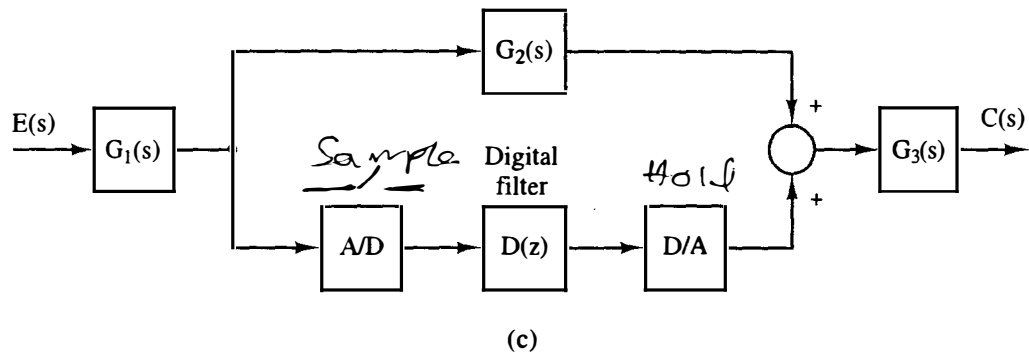


Figure P4-10 Systems for Problem 4-10.

- 4-11. Example 4.3 calculates the step response of the system in Figure 4-2. Example 4.4 calculates the step response of the same system preceded by a digital filter with the transfer function $D(z) = (2 - z^{-1})$. This system is shown in Figure P4-11.
- Solve for the output of the digital filter $m(kT)$.
 - Let the response in Example 4.3 be denoted as $c_1(kT)$. Use the results in part (a) to express the output $c(kT)$ in Figure P4-11 as a function of $c_1(kT)$.
 - Use the response $c_1(kT)$ calculated in Example 4.3 and the results in part (b) to find the output $c(kT)$ in Figure P4-11. This result should be the same as in Example 4.4.
 - Use the response $C_1(z)$ calculated in Example 4.3 and the result in part (b) to find the output $C(z)$ in Figure P4-11. This result should be the same as in Example 4.4.

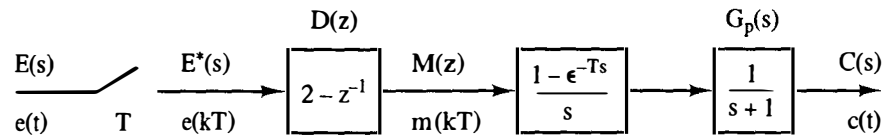


Figure P4-11 System for Problem 4-11.

- 4-12. Consider the *hardware* depicted in Figure P4-12. The transfer function of the digital controller implemented in the computer is given by

$$D(z) = \frac{4.5(z - 0.90)}{z - 0.85}$$

The input voltage rating of the analog-to-digital converter is ± 10 V and the output voltage rating of the digital-to-analog converter is also ± 10 V.

- Calculate the dc gain of the controller.
- State exactly how you would verify, using the hardware, the value calculated in part (a). Give the equipment required, the required settings on the equipment, and the expected measurements.

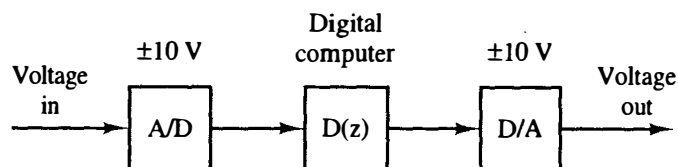


Figure P4-12 Hardware configuration for Problem 4-12.

4-13. For the system of Figure P4-13, the filter solves the difference equation

$$m(k) = 0.9m(k-1) + 0.2e(k)$$

The sampling rate is 1 Hz and the plant transfer function is given by

$$G_p(s) = \frac{1}{s + 0.2}$$

- Find the system transfer function $C(z)/E(z)$.
- Find the system dc gain from the results of part (a).
- Verify the results of part (b) by finding the dc gain of the filter using $D(z)$ and that of the plant using $G_p(s)$.
- Use the results of part (b) to find the steady-state value of the unit-step response.
- Verify the results of part (d) by calculating $c(kT)$ for a unit-step input.
- Note that in part (e), the coefficients in the partial-fraction expansion add to zero. Why does this occur?

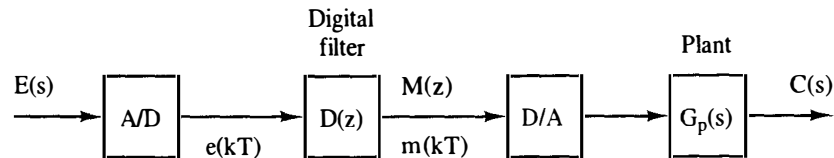


Figure P4-13 System for Problems 4-13 and 4-14.

4-14. Repeat Problem 4-13 for the case that the filter solves the difference equation

$$m(k+1) = 0.5e(k+1) - (0.5)(0.98)e(k) + 0.995m(k)$$

the sampling rate is 10 Hz, and the plant transfer function is given by

$$G_p(s) = \frac{5}{(s+1)(s+2)}$$

4-15. Shown in Figure P4-15 is the block diagram of one joint of a robot arm. This system is described in Problem 1-16. The signal $M(s)$ is the sampler input, $E_a(s)$ is the servomotor input voltage, $\theta_m(s)$ is the motor shaft angle, and the output $\theta_a(s)$ is the angle of the arm.

- Suppose that the sampling-and-data-reconstruction process is implemented with an analog-to-digital converter (A/D) and a digital-to-analog converter. Redraw Figure P4-15 showing the A/D and the D/A.
- Suppose that the units of $e_a(t)$ are volts, and of $\theta_m(t)$ are rpm. The servomotor is

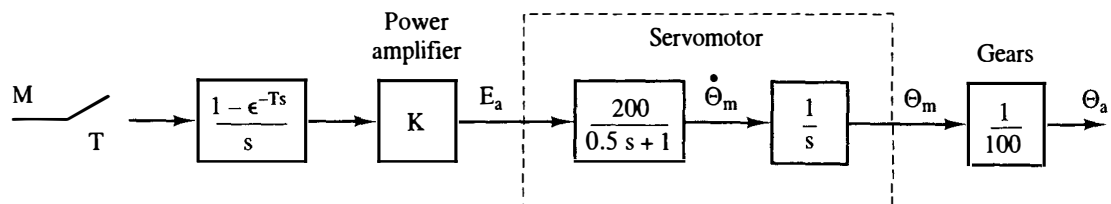


Figure P4-15 Model of robot arm joint.

rated at 24 V (the voltage $e_a(t)$ should be less than or equal to 24 V in magnitude). Commercially available D/As are usually rated with output voltage ranges of ± 5 V, ± 10 V, 0 to 5 V, 0 to 10 V, or 0 to 20 V. If the gain of the power amplifier is 2.4, what should be the rated voltage of the D/A? Why?

- (c) Derive the analog transfer function $\theta_a(s)/E_a(s)$.
- (d) With $K = 2.4$ and $T = 0.1$ s, derive the pulse transfer function $\theta_a(z)/M(z)$.
- (e) Derive the steady-state output for $m(t)$ constant. Justify this value from the motor characteristics.
- (f) Verify the results of part (d) by computer.

4-16. Figure P4-16 illustrates a thermal stress chamber. This system is described in Problem 1-10. The system output $c(t)$ is the chamber temperature in degrees Celsius, and the control-voltage input $m(t)$ operates a valve on a steam line. The sensor is based on a thermistor, which is a temperature-sensitive resistor. The disturbance input $d(t)$ models the opening of the door into the chamber. With the door closed, $d(t) = 0$; if the door is opened at $t = t_0$, $d(t) = u(t - t_0)$, a unit-step function.

- (a) Suppose that the sampling-and-data-reconstruction process is implemented with an analog-to-digital converter (A/D) and a digital-to-analog converter (D/A). Redraw Figure P4-16 showing the A/D and the D/A.
- (b) Derive the transfer function $C(z)/E(z)$.
- (c) A constant voltage of $e(t) = 10$ V is applied for a long period of time. Find the steady-state temperature of the chamber, with the door closed. Note that this problem can be solved without knowing the sample period T .
- (d) Find the steady-state effect on the chamber temperature of leaving the door open.
- (e) Find the expression for $C(s)$ as a function of the Laplace-transform variable s , the control input, and the disturbance input. The z -transform variable z cannot appear in this expression.

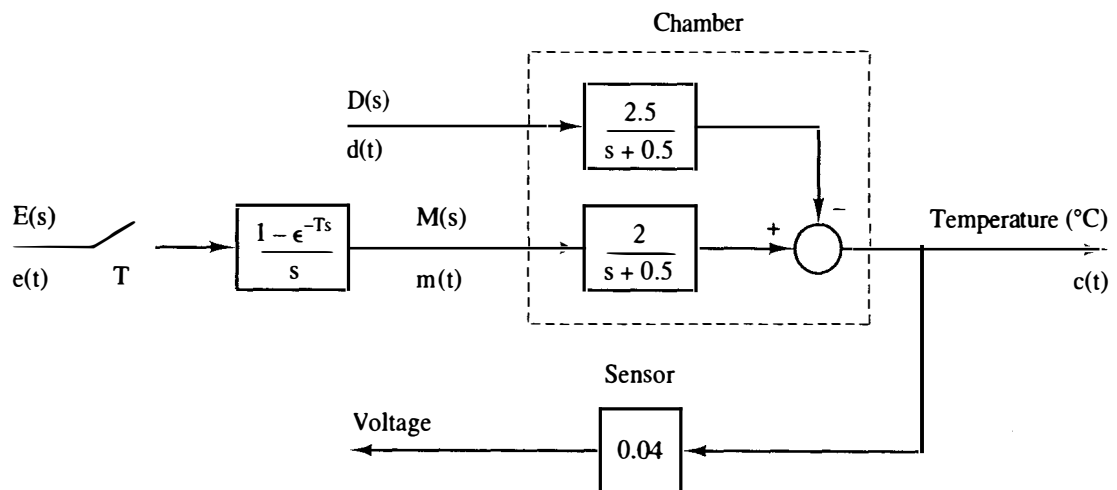


Figure P4-16 Block diagram for a thermal test chamber.

4-17. Given in Figure P4-17 is the block diagram of a rigid-body satellite. The control signal is the voltage $e(t)$. The zero-order hold output $m(t)$ is converted into a torque $\tau(t)$ by an amplifier and the thrusters (see Section 1.4). The system output is the attitude angle $\theta(t)$ of the satellite.

- (a) Find the transfer function $\Theta(z)/E(z)$.

- (b) Use the results of part (a) to find the system's unit-step response, that is, the response with $e(t) = u(t)$.
- (c) Sketch the zero-order-hold output $m(t)$ in (b).
- (d) Use $m(t)$ in part (c) to find $c(t) = \mathcal{L}^{-1}[KM(s)/Js^2]$.
- (e) In part (d), evaluate $c(kT)$. This response should equal that found in part (b).

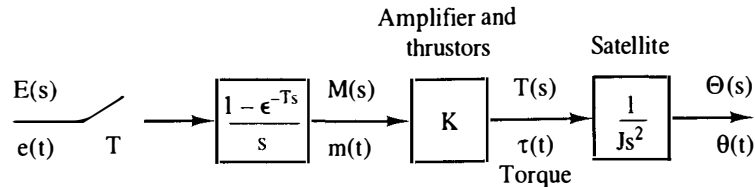


Figure P4-17 Block diagram for a satellite.

- 4-18.** The antenna positioning system described in Section 1.5 and Problem 1-7 is depicted in Figure P4-18. In this problem we consider the yaw angle control system, where $\theta(t)$ is the yaw angle. The angle sensor (a digital shaft encoder and the data hold) yields $v_o(kT) = [0.4\theta(kT)]$, where the units of $v_o(t)$ are volts and $\theta(t)$ are degrees. The sample period is $T = 0.05$ s.
- (a) Find the transfer function $\Theta(z)/E(z)$.
- (b) The yaw angle is initially zero. The input voltage $e(t)$ is set equal to 10 V at $t = 0$, and is zero at each sample period thereafter. Find the steady-state value of the yaw angle.
- (c) Note that in part (b), the coefficients in the partial-fraction expansion add to zero. Why does this occur?
- (d) The input voltage $e(t)$ is set to a constant value. Without solving mathematically, give a description of the system response.
- (e) Suppose in part (d) that you are observing the antenna. Describe what you would see.
- (f) Verify the results of part (a) by computer.

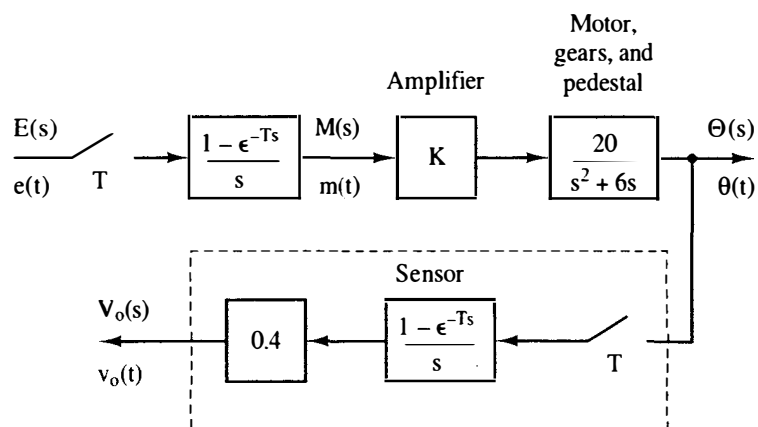


Figure P4-18 Block diagram for an antenna control system.

- 4-19.** Find the modified z -transform of the following functions.

(a) $E(s) = \frac{20}{(s+2)(s+5)}$

(b) $E(s) = \frac{5}{s(s+1)}$

$$(c) E(s) = \frac{s+2}{s(s+1)}$$

$$(d) E(s) = \frac{s+2}{s^2(s+1)}$$

$$(e) E(s) = \frac{s^2+5s+6}{s(s+4)(s+5)}$$

$$(f) E(s) = \frac{2}{s^2+2s+5}$$

4-20. Find the z -transform of the following functions. The results of Problem 4-19 may be useful.

$$(a) E(s) = \frac{20e^{-0.3Ts}}{(s+2)(s+5)}$$

$$(b) E(s) = \frac{5e^{-0.6Ts}}{s(s+1)}$$

$$(c) E(s) = \frac{(s+2)e^{-1.1Ts}}{s(s+1)}$$

$$(d) E(s) = \frac{(s+2)e^{-0.2Ts}}{s^2(s+1)}$$

$$(e) E(s) = \frac{(s^2+5s+6)e^{-0.3Ts}}{s(s+4)(s+5)}$$

$$(f) E(s) = \frac{2e^{-0.75Ts}}{s^2+2s+5}$$

4-21. Generally, a temperature control system is modeled more accurately if an ideal time delay is added to the plant. Suppose that in the thermal test chamber of Problem 4-16, the plant transfer function is given by

$$G_p(s) = \frac{C(s)}{E(s)} = \frac{2e^{-2s}}{s+0.5}$$

Hence the plant has a 2-s time delay before its response to an input. For this problem, let the sample period $T = 0.6$ s.

- Find the unit step response for the system of Figure P4-16; that is, find $c(kT)$ with $e(t) = u(t)$ and $d(t) = 0$, and with no delay.
- Repeat part (a), with the 2-s time delay included in $G_p(s)$, as given above.
- Solve for $c(t)$ with no delay, using the Laplace transform and the plant input $m(t)$.
- Find $c(t)$ for the delay included, using the results of part (c).
- Show that in part (c), $c(t)$ evaluated at $t = kT$ yields $c(kT)$ in part (a).
- Show that in part (d), $c(t)$ evaluated at $t = kT$ yields $c(kT)$ in part (b).

4-22. For the thermal test chamber of Problems 4-16 and 4-21, let $T = 0.6$ s. Suppose that a proportional-integral (PI) digital controller with the transfer function

$$D(z) = 1.2 + \frac{0.1z}{z-1}$$

is inserted between the sampler and the zero-order hold in Figure P4-16.

- With $d(t) = 0$ and $e(t) = u(t)$, solve for $c(kT)$, with the time delay of 2 s omitted.
 - Explain what happens to the temperature in the chamber in part (a). Is this result physically possible?
 - In Figure P4-16, calculate $m(kT)$, the signal that controls the valve in the steam line, for the inputs of part (a).
 - Considering the physical characteristics of a valve, what happens to the temperature in the physical test chamber?
- 4-23.** Consider the system of Figure P4-23. The plant is described by the first-order differential equation

$$\frac{dy(t)}{dt} + 0.05y(t) = 0.1m(t)$$

Let $T = 2$ s.

- (a) Find the system transfer function $Y(z)/E(z)$.
- (b) Draw a discrete simulation diagram, using the results of part (a), and give the state equations for this diagram.
- (c) Draw a continuous-time simulation diagram for $G_p(s)$, and give the state equations for this diagram.
- (d) Use the state-variable model of part (c) to find a discrete state model for the system. The state vectors of the discrete system and the continuous system are to be the same.
- (e) Draw a simulation diagram for the discrete state model in part (d).
- (f) Use Mason's gain formula to find the transfer function in part (e), which must be the same as found in part (a).
- (g) Verify the results in parts (a) and (d) by computer.

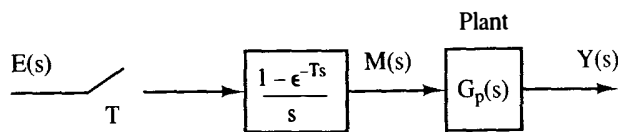


Figure P4-23 System for Problem 4-23.

- 4-24. Repeat Problem 4-23 for the plant described by the second-order differential equation

$$\frac{d^2 y(t)}{dt^2} + 0.15 \frac{dy(t)}{dt} + 0.005 y(t) = 0.1 m(t)$$

- 4-25. Consider the robot arm system of Figure P4-15. Let $T = 0.1$ s.

- (a) Find the system transfer function $\Theta_a(z)/M(z)$.
- (b) Draw a discrete simulation diagram, using the results of part (a), and give the state equations for this diagram.
- (c) Draw a continuous-time simulation diagram for amplifier-servomotor-gears system, and give the state equations for this diagram.
- (d) Use the state-variable model of part (c) to find a discrete state model for the system. The states of the discrete systems are the same as those of the continuous system.
- (e) Draw a simulation diagram for the discrete state model in part (d).
- (f) Use Mason's gain formula to find the transfer function in part (e), which must be the same as found in part (a).
- (g) Verify the results in parts (a) and (d) by computer.

- 4-26. Consider the thermal stress chamber depicted in Figure P4-16. Let $T = 0.6$ s. Ignore the disturbance input $d(t)$ for this problem.

- (a) Find the system transfer function $C(z)/E(z)$.
- (b) Draw a discrete simulation diagram, using the results of part (a), and give the state equations for this diagram.
- (c) Draw a continuous-time simulation diagram for the plant, and give the state equations for this diagram.
- (d) Use the state-variable model of part (c) to find a discrete state model for the system. The states of the discrete systems are the same as those of the continuous system.
- (e) Draw a simulation diagram for the discrete state model in part (d).
- (f) Use Mason's gain formula to find the transfer function in part (e), which must be the same as that found in part (a).
- (g) Verify the results in parts (a) and (d) by computer.

- 4-27. Repeat Problem 4-26 for the satellite system of Figure P4-17, with $T = 1$ s. In part (a), the required transfer function is $\Theta(z)/E(z)$.
- 4-28. Repeat Problem 4-26 for the antenna system of Figure P4-18, with $T = 0.05$ s. In part (a), the required transfer function is $\Theta(z)/E(z)$.
- 4-29. Consider a proportional-integral (PI) digital controller with the transfer function

$$D(z) = 1.2 + \frac{0.1z}{z - 1}$$

This controller is placed in the designated system between the sampler and data hold. Find a discrete state model of:

- (a) The system of Problem 4-23.
- (b) The satellite system of Problem 4-27.
- (c) In parts (a) and (b), use computer computations to verify the results by calculating the system transfer functions from the state models.
- 4-30. Find a discrete-state-variable representation for the system shown in Figure P4-30. A discrete-state-variable description of the continuous system is given by

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.9 & 1 & 2 \\ 0 & 0.9 & 0 \\ -1 & 0 & 0.5 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} m(k)$$

$$y(k) = [1 \quad 1.5 \quad 2.3] \mathbf{x}(k)$$

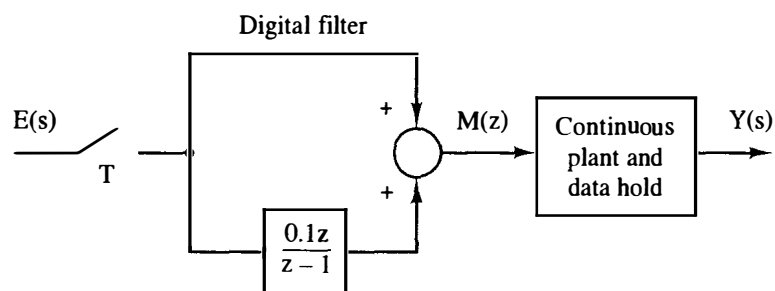


Figure P4-30 System for Problem 4-30.

- 4-31. Consider the system of Figure P4-31. The filter transfer function is $D(z)$.
- (a) Express $C(z)$ as a function of E .
- (b) A discrete state model of this system does not exist. Why?
- (c) What assumptions concerning $e(t)$ must be made in order to derive an approximate discrete state model?

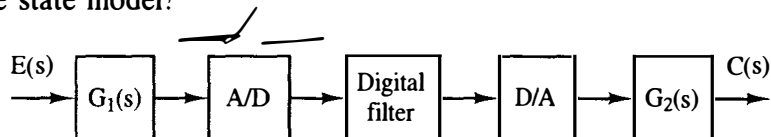


Figure P4-31 System for Problem 4-31.

- 4-32. (a) The model of a continuous system with algebraic loops is given as

$$\dot{x}(t) = -2x(t) + 0.5\dot{x}(t) + 3u(t)$$

$$y(t) = \dot{x}(t) + 4u(t)$$

Derive the state equations for this system.

(b) Repeat part (a) for the system described by

$$\dot{x}_1(t) = -x_1(t) + 2\dot{x}_2(t) + u_1(t)$$

$$\dot{x}_2(t) = -\dot{x}_2(t) - x_2(t) - \dot{x}_1(t) + x_1(t) + u_2(t)$$

$$y(t) = \dot{x}_2(t)$$

Use the matrix technique of Section 4.9.

(c) Verify the results of part (b) by solving the given equations for $\dot{x}_1(t)$ and $\dot{x}_2(t)$ in the standard state-variable format.

Closed-Loop Systems

5.1 INTRODUCTION

In Chapter 4 a technique was developed for determining the output functions of open-loop discrete linear time-invariant (LTI) systems. In this chapter we extend these techniques to determine the output functions of closed-loop discrete LTI systems. Also presented is a technique for developing state-variable models for closed-loop discrete systems.

5.2 PRELIMINARY CONCEPTS

Before considering simple closed-loop systems, open-loop systems with cascaded plants will be reviewed. As shown in Chapter 4, the output for the system of Figure 5-1a is

$$C(z) = G_1(z)G_2(z)E(z) \quad (5-1)$$

and that of Figure 5-1b is

$$C(z) = \overline{G_1} G_2(z)E(z) \quad (5-2)$$

For the system of Figure 5-1c,

$$C(s) = G_2(s)A^*(s) = G_2(s)\overline{G_1} E^*(s) \quad (5-3)$$

and

$$C(z) = G_2(z)\overline{G_1} E(z) \quad (5-4)$$

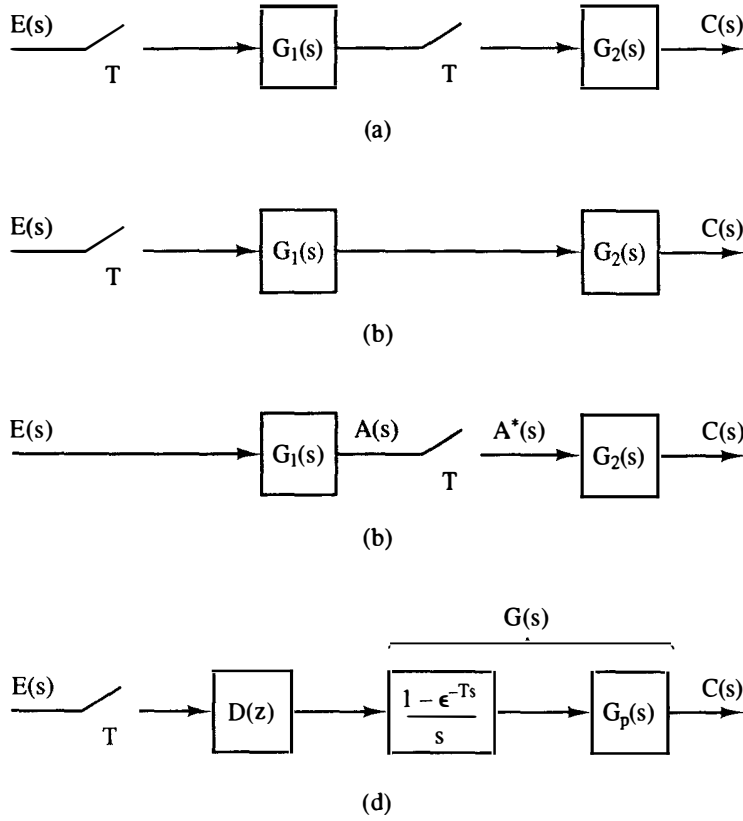


Figure 5-1 Open-loop sampled-data systems.

For this case, no transfer function can be found since $E(z)$ cannot be factored from $\overline{G_1 E}(z)$. In general, no transfer function can be written for the system in which the input is applied to an analog element before being sampled. However, the output can always be expressed as a function of the input, and as will be shown later, this type of system presents no particular difficulties in either analysis or design. For the system of Figure 5-1d, the output is

$$C(z) = D(z)G(z)E(z)$$

where

$$G(z) = \mathcal{Z}\left[\frac{1 - e^{-Ts}}{s} G_p(s)\right] = \frac{z - 1}{z} \mathcal{Z}\left[\frac{G_p(s)}{s}\right]$$

We now derive the output function for the system of Figure 5-2. First we write

$$C(s) = G(s)E^*(s) \quad (5-5)$$

and

$$E(s) = R(s) - H(s)C(s) \quad (5-6)$$

Substituting (5-5) into (5-6), we obtain

$$E(s) = R(s) - G(s)H(s)E^*(s) \quad (5-7)$$

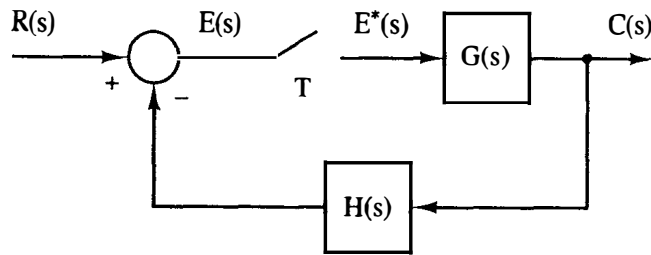


Figure 5-2 Closed-loop sampled-data system.

and by taking the starred transform (see Section 4.3), we have

$$E^*(s) = R^*(s) - \overline{GH}^*(s)E^*(s) \quad (5-8)$$

Solving for $E^*(s)$, we obtain

$$E^*(s) = \frac{R^*(s)}{1 + \overline{GH}^*(s)} \quad (5-9)$$

and from (5-5),

$$C(s) = G(s) \frac{R^*(s)}{1 + \overline{GH}^*(s)} \quad (5-10)$$

which yields an expression for the continuous output. The sampled output is, then,

$$C^*(s) = G^*(s)E^*(s) = \frac{G^*(s)R^*(s)}{1 + \overline{GH}^*(s)}, \quad C(z) = \frac{G(z)R(z)}{1 + \overline{GH}(z)} \quad (5-11)$$

Problems can be encountered in deriving the output function of a closed-loop system. This can be illustrated for the case above. Suppose that (5-6) had been starred and substituted into (5-5). Then

$$C(s) = G(s)R^*(s) - G(s)\overline{HC}^*(s) \quad (5-12)$$

and $C^*(s)$ is

$$C^*(s) = G^*(s)R^*(s) - G^*(s)\overline{HC}^*(s) \quad (5-13)$$

In general, $C^*(s)$ cannot be factored from $\overline{HC}^*(s)$. Thus (5-13) cannot be solved for $C^*(s)$. In general, in analyzing a system, an equation should not be starred if a system signal is lost as a factor, as shown above. However, for systems more complex than the one above, solving the system equations can become quite complex. A simpler method of analysis will now be developed that avoids this problem.

First, however, consider the system of Figure 5-3. Since the input is not sampled before being applied to an analog element, no transfer function can be derived.

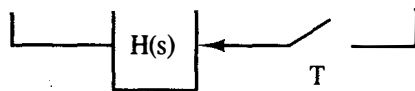


Figure 5-3 Sampled-data system.

Nevertheless, the system can be analyzed. From Figure 5-3 we note that

$$C(s) = G(s)E(s) \quad (5-14)$$

and

$$E(s) = R(s) - H(s)C^*(s) \quad (5-15)$$

Substituting (5-15) into (5-14), we obtain

$$C(s) = G(s)R(s) - G(s)H(s)C^*(s) \quad (5-16)$$

Starring (5-16) yields

$$C^*(s) = \overline{GR}^*(s) - \overline{GH}^*(s)C^*(s) \quad (5-17)$$

In this system, the forcing function $R(s)$ is necessarily lost as a factor in (5-17), since $r(t)$ is not sampled. Solving (5-17) for $C^*(s)$, we obtain

$$C^*(s) = \frac{\overline{GR}^*(s)}{1 + \overline{GH}^*(s)}, \quad C(z) = \frac{\overline{GR}(z)}{1 + \overline{GH}(z)} \quad (5-18)$$

The continuous output is obtained from (5-16) and (5-18) as

$$C(s) = G(s)R(s) - \frac{G(s)H(s)\overline{GR}^*(s)}{1 + \overline{GH}^*(s)} \quad (5-19)$$

For the system of Figure 5-3, then, no transfer function can be derived; the problem is that the input is not sampled before being applied to an analog part of the system. In a practical system, we generally have more than one input. Consider, as an example, an aircraft flying in a closed-loop mode (i.e., flying on autopilot). Suppose also that the autopilot is implemented digitally. Thus this system is a closed-loop digital control system. The commands into this system (e.g., an altitude change) may be inputted into the control computer through an analog-to-digital converter, or may even be generated by the computer itself. Thus this input is sampled, and a transfer function may be developed. However, the vertical component of the wind, which must also be considered as an input into the altitude control system, is not sampled. Hence a transfer function can be derived from the command input to the output (the altitude); however, a transfer function cannot be developed from wind input to aircraft altitude as the output.

5.3 DERIVATION PROCEDURE

The determination of transfer functions for sampled-data systems is difficult because a transfer function for the ideal sampler does not exist. A procedure for finding transfer functions will now be developed using the system of Figure 5-4, which has the flow graph shown in Figure 5-5. We omit the sampler from the system signal flow graph, since a transfer function cannot be written for the device. Note that the effect of the sampler is included in the flow graph, since the sampler output is shown in

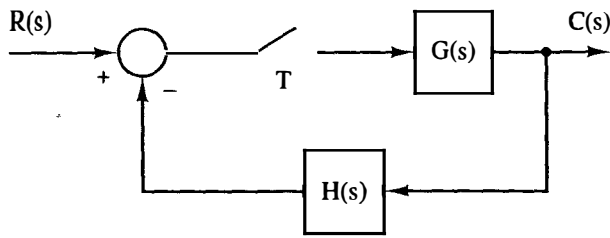


Figure 5-4 Sampled-data control system.

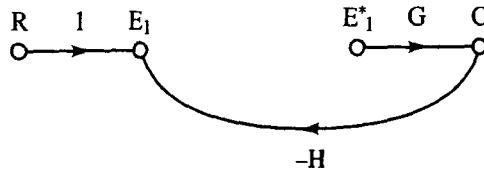


Figure 5-5 Original flow graph for the sampled-data system in Figure 5-4.

starred form, E_1^* , and E_1^* will be treated as an input. This flow graph is referred to as the original signal flow graph. The sampled output $C^*(s)$ can be found for a discrete-time system of this type by employing the following step-by-step procedure.

1. Construct the *original signal flow graph*. This has been done for the system and is shown in Figure 5-5.
2. Assign a variable to each sampler input. Then the sampler output is this variable starred. For this example system E_1 is the input to the sampler and E_1^* is the sampler output.
3. Considering each sampler output to be a source node (input), express the sampler inputs and the system output in terms of each sampler output (which is treated as an input in the flow graph), and the system input. For this example,

$$E_1 = R - GHE_1^* \quad (5-20)$$

$$C = GE_1^* \quad (5-21)$$

where $E_1 = E_1(s)$, and so on. For convenience, the dependency on s will not be shown.

4. Take the starred transform of these equations and solve by any convenient method. For the example,

$$E_1^* = R^* - \overline{GH}^* E_1^* \quad (5-22)$$

$$C^* = G^* E_1^* \quad (5-23)$$

From (5-22),

$$E_1^* = \frac{R^*}{1 + \overline{GH}^*} \quad (5-24)$$

and, from (5-23) and (5-24),

$$C^*(s) = \frac{G^*(s)}{1 + \overline{GH}^*(s)} R^*(s) \quad (5-25)$$

If there is more than one sampler in the system, the equations for the starred variables may be solved using Cramer's rule, or any other technique that is applicable to the solution of linear simultaneous equations.

The systems equations can also be solved by constructing a signal flow graph from these equations and applying Mason's gain formula. This flow graph is called the *sampled signal flow graph*. This method is sometimes superior to Cramer's rule, provided that the sampled signal flow graph is simple enough so that all loops are easily identified. To illustrate this approach, consider again equations (5-21), (5-22), and (5-23). The signal flow graph for these equations is shown in Figure 5-6. From the flow graph,

$$C^*(s) = \frac{G^*(s)}{1 + \overline{GH}^*(s)} R^*(s) \quad (5-26)$$

and this result agrees with (5-25). Note also from Figure 5-6 that

$$C(s) = \frac{G(s)}{1 + \overline{GH}^*(s)} R^*(s) \quad (5-27)$$

This method for finding the output function for closed-loop systems will now be illustrated via the following examples.

Example 5.1

Consider the digital control system of Figure 5-7a which has the model given in Figure 5-7b (see Section 4.4). The original flow graph is shown in Figure 5-8. We can write

$$E = R - GHD^*E^*$$

$$C = GD^*E^*$$

Hence

$$E^* = R^* - \overline{GH}^*D^*E^* \rightarrow E^* = \frac{R^*}{1 + D^*\overline{GH}^*}$$

$$C^* = G^*D^*E^*$$

Thus

$$C(z) = \frac{D(z)G(z)}{1 + D(z)\overline{GH}(z)} R(z)$$

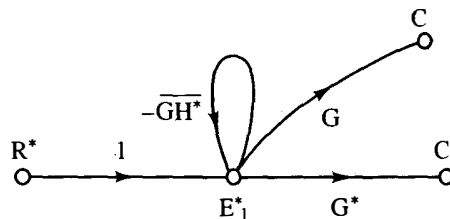


Figure 5-6 Sampled signal flow graph for the system in Figure 5-4.

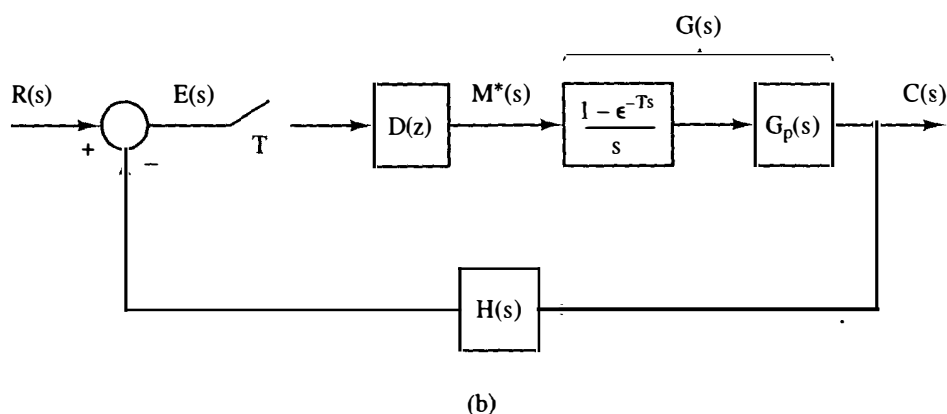
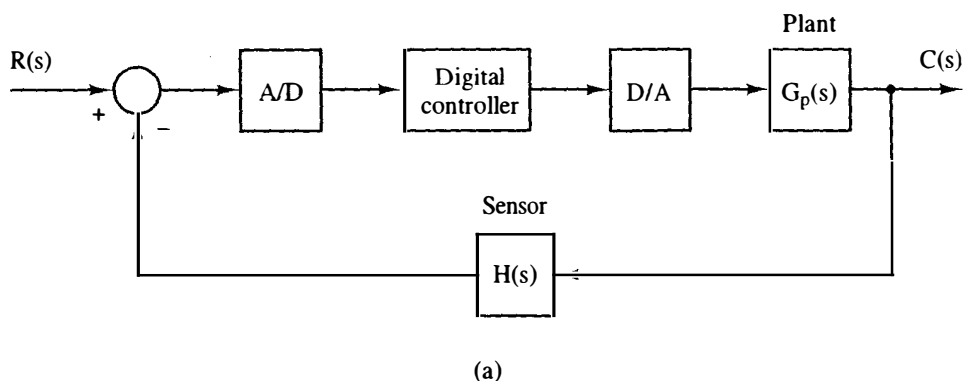


Figure 5-7 Closed-loop digital control system.

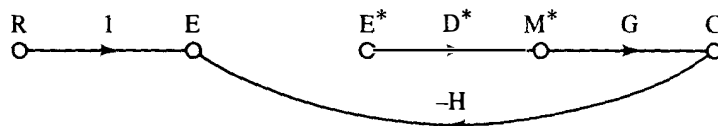


Figure 5-8 Original flow graph for Example 5.1.

Example 5.2

Consider the system shown in Figure 5-9. The original signal flow graph is shown in Figure 5-10. The system equations are

$$E_1 = R - G_2 E_2^*$$

$$E_2 = G_1 E_1^* - G_2 H E_2^*$$

$$C = G_2 E_2^*$$

Starring these equations, we obtain

$$E_1^* = R^* - G_2^* E_2^*$$

$$E_2^* = G_1^* E_1^* - \overline{G_2 H}^* E_2^*$$

$$C^* = G_2^* E_2^*$$

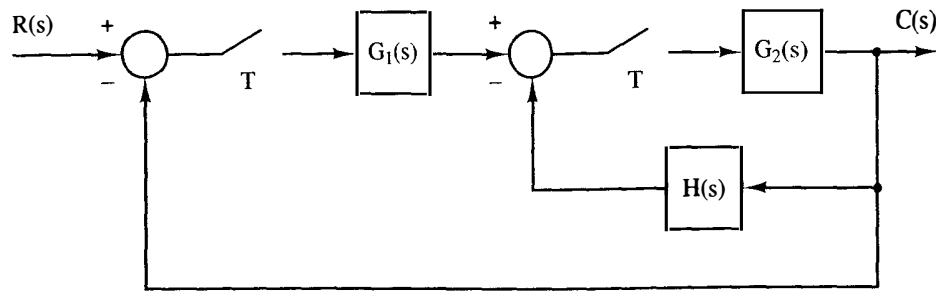


Figure 5-9 Sampled-data system for Example 5.2.

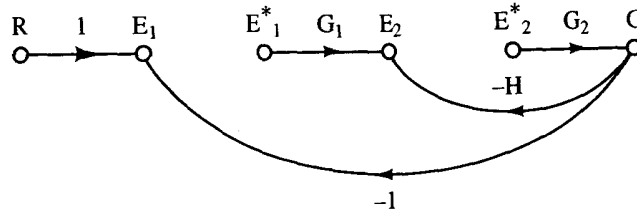


Figure 5-10 Original flow graph for Example 5.2.

The sampled flow graph can then be drawn from these equations as shown in Figure 5-11. Then applying Mason's gain formula, we obtain

$$C^* = \frac{G_1^* G_2^*}{1 + G_1^* G_2^* + \overline{G_2 H^*}} R^* \quad \text{or} \quad C(z) = \frac{G_1(z) G_2(z) R(z)}{1 + G_1(z) G_2(z) + \overline{G_2 H}(z)}$$

and

$$C(s) = \frac{G_2(s) G_1^*(s)}{1 + G_1^*(s) G_2^*(s) + \overline{G_2 H^*}(s)} R^*(s)$$

Example 5.3

As another example, consider the system shown in Figure 5-12. Note that no transfer function may be derived for this system, since the input $R(s)$ reaches $G_2(s)$ without being sampled.

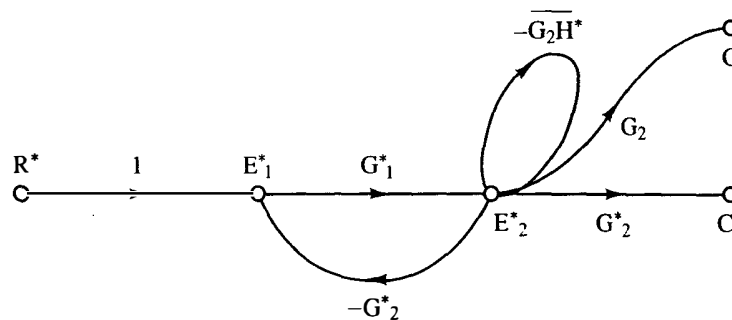


Figure 5-11 Sampled flow graph for Example 5.2.

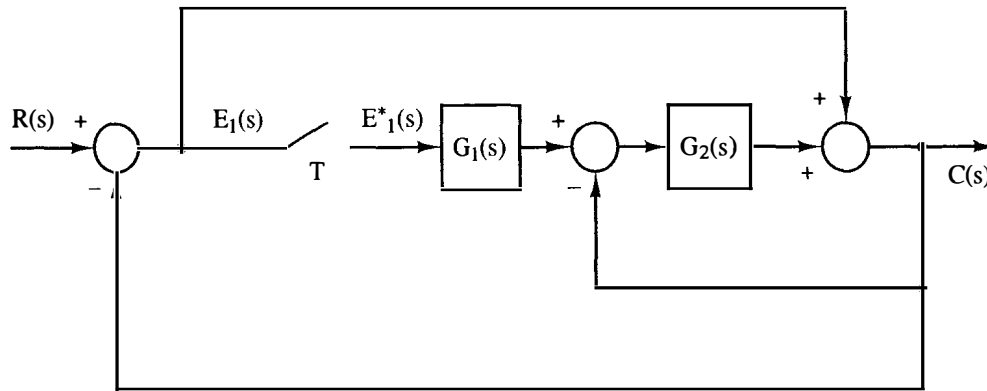


Figure 5-12 Sampled-data system for Example 5.3.

The original signal flow graph is given in Figure 5-13. From this figure note that

$$\begin{aligned} E_1 &= R - C \\ C &= E_1 + G_2[G_1 E_1^* - C] \\ &= R - C + G_1 G_2 E_1^* - G_2 C \end{aligned}$$

Therefore,

$$[1 + 1 + G_2]C = R + G_1 G_2 E_1^*$$

and

$$\begin{aligned} C &= \frac{R}{2 + G_2} + \frac{G_1 G_2}{2 + G_2} E_1^* \\ E_1 &= R - C = \frac{(1 + G_2)R}{2 + G_2} - \frac{G_1 G_2}{2 + G_2} E_1^* \end{aligned}$$

Or the equation for E_1 and C can be written directly from Figure 5-13 using Mason's formula. Hence

$$\begin{aligned} C^* &= \left[\frac{R}{2 + G_2} \right]^* + \left[\frac{G_1 G_2}{2 + G_2} \right]^* E_1^* \\ E_1^* &= \left[\frac{(1 + G_2)R}{2 + G_2} \right]^* - \left[\frac{G_1 G_2}{2 + G_2} \right]^* E_1^* \end{aligned}$$

The sampled flow graph derived from these equations is given in Figure 5-14.

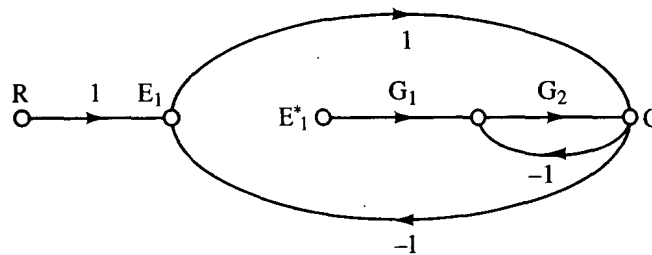


Figure 5-13 Original flow graph for Example 5.3.

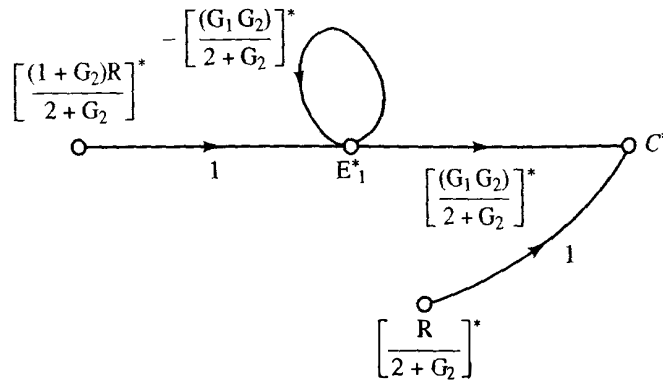


Figure 5-14 Sampled flow graph for Example 5.3.

Then by employing Mason's gain formula, we obtain

$$C^*(s) = \left[\frac{R}{2+G_2} \right]^*(s) + \frac{\left[\frac{G_1 G_2}{2+G_2} \right]^*(s)}{1 + \left[\frac{G_1 G_2}{2+G_2} \right]^*(s)} \left[\frac{(1+G_2)R}{2+G_2} \right]^*(s)$$

Note that, as stated above, no transfer function is possible, since the input is fed into a continuous element in the system, $G_2(s)$, without first being sampled.

As mentioned above, matrix methods, or Cramer's rule, may be simpler to use in solving the system equations, if there are many loops in the sampled signal flow graph. This is especially true when all the loops of the flow graph are not easily identified. However, once one has gained experience and proficiency in the use of signal flow graphs, these flow graphs can be easily used to obtain quick and accurate solutions for less complex systems.

Example 5.4

As a final example, consider the system of Figure 5-15, which contains a digital controller with a computation time of t_0 seconds, $t_0 < T$. The effect of the computation time is modeled as the ideal time delay (see Section 4.6).

The original signal flow graph is given in Figure 5-16. The system equation is then, from Mason's gain formula,

$$C = \frac{GR}{1 + GH_1} - \frac{GH_2 \epsilon^{-t_0 s}}{1 + GH_1} D^* C^*$$

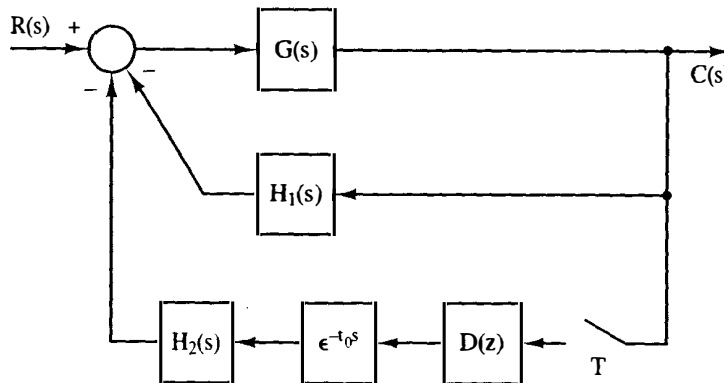


Figure 5-15 System for Example 5.4.

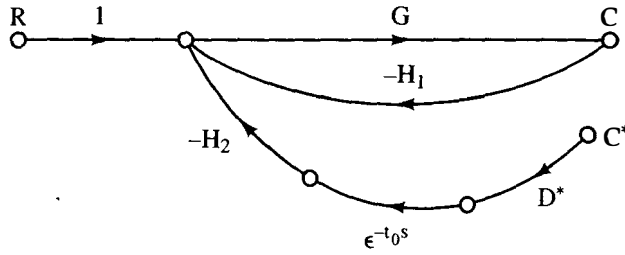


Figure 5-16 Original flow graph for Example 5.4.

and thus

$$C(z) = \left[\frac{GR}{1 + GH_1} \right](z) - \left[\frac{GH_2}{1 + GH_1} \right](z, m)D(z)C(z)$$

where $mT = T - t_0$. The second equation may be solved directly for $C(z)$.

$$C(z) = \frac{\left[\frac{GR}{1 + GH_1} \right](z)}{1 + \left[\frac{GH_2}{1 + GH_1} \right](z, m)D(z)}$$

If the computational delay is greater than one sampling interval (i.e., if $t_0 = kT + \Delta T$, k is a positive integer), the denominator becomes

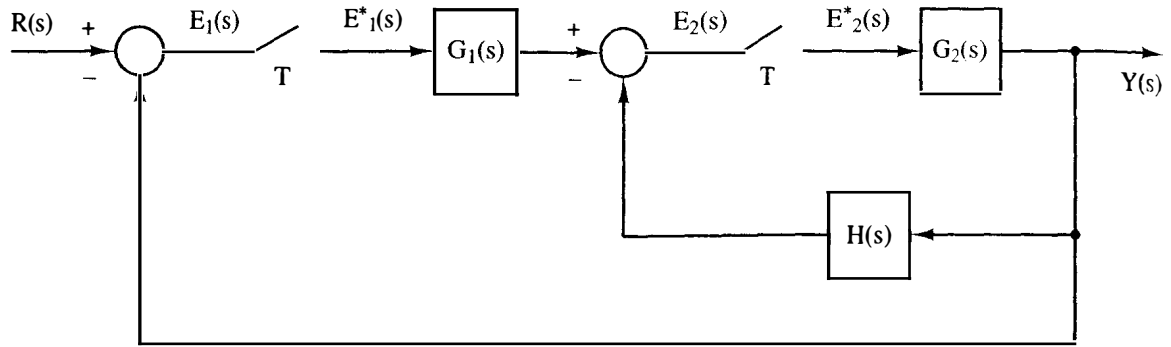
$$1 + z^{-k} \left[\frac{GH_2}{1 + GH_1} \right](z, m)D(z)$$

where $mT = T - \Delta T$.

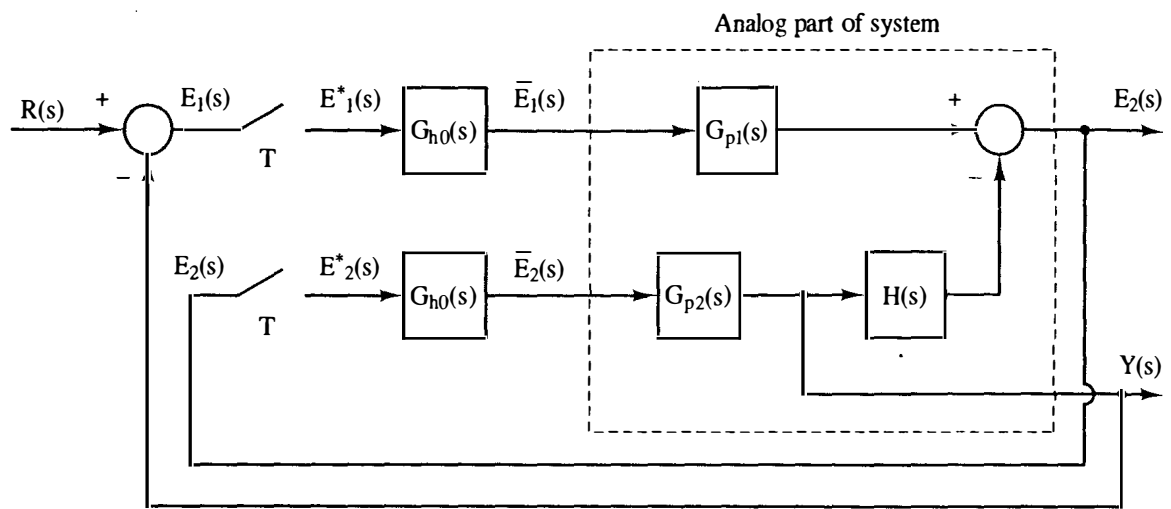
5.4 STATE-VARIABLE MODELS

A technique for finding the transfer function of a closed-loop discrete-time system was presented above. As shown in Chapter 4, a discrete-state-variable model may be generated directly from the transfer function. However, this technique has the disadvantage that the analog system physical variables generally do not appear as discrete state variables.

The technique of converting continuous state equations to discrete state equations, presented in Section 4.10, may also be utilized in determining a discrete-state-variable model for a closed-loop system. The application of this technique will be illustrated by an example. Consider the system of Example 5.2, which is repeated in Figure 5-17a. As a first step, the system is redrawn such that zero-order-hold outputs are shown as inputs, and sampler inputs and the system output are shown as outputs. The results of this step are shown in Figure 5-17b. Hence we are considering the analog part of the system as in Figure 5-18a [$E_1(s)$ is not shown as an output, since it is determined directly from the system output $Y(s)$]. Next the continuous state equations for this part of the system are written, and from these equations, the discrete state equations are generated. For this system, the discrete



(a)



(b)

Figure 5-17 Closed-loop system.

state equations will be of the form

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbf{A}_1 \mathbf{v}(k) + \mathbf{B}_1 \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix} \\ \begin{bmatrix} y(k) \\ e_2(k) \end{bmatrix} &= \mathbf{C}_1 \mathbf{v}(k) + \mathbf{D}_1 \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix} \end{aligned} \quad (5-28)$$

Either a discrete simulation diagram or a flow graph is then constructed from these state equations, and should include all connecting paths of the closed-loop system external to the simulation diagram for (5-28). The result for the system considered is shown in Figure 5-18b. From this simulation diagram the system discrete state equations may be written. An example to illustrate this technique will now be given.

Example 5.5

The discrete state model for the system of Figure 5-17 will be derived, with $T = 0.1$ s, and with

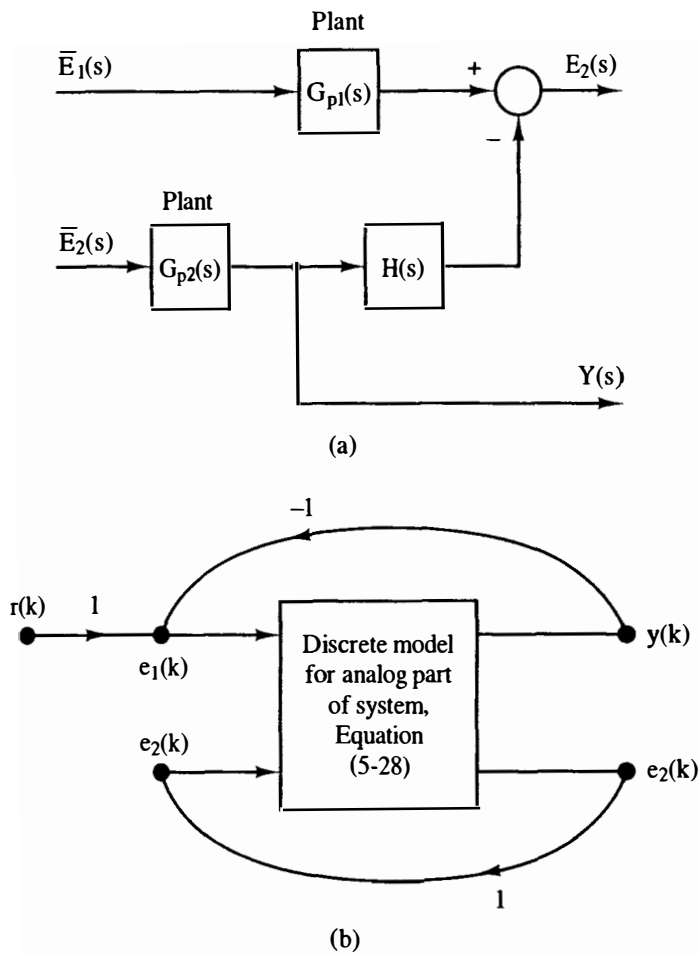


Figure 5-18 Technique for determining the discrete state model.

$$G_1(s) = \frac{1 - \epsilon^{-Ts}}{s^2(s+1)} = \frac{1 - \epsilon^{-Ts}}{s} G_{p1}(s)$$

$$G_2(s) = \frac{2(1 - \epsilon^{-Ts})}{s(s+2)} = \frac{1 - \epsilon^{-Ts}}{s} G_{p2}(s)$$

$$G_{p1}(s) = \frac{1}{s(s+1)}, \quad G_{p2}(s) = \frac{2}{s+2}, \quad H(s) = \frac{10}{s+10}$$

A flow graph of the system of Figure 5.18a is shown in Figure 5-19a. From this flow graph we write the continuous state equations:

$$\dot{\mathbf{v}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -10 & 10 \\ 0 & 0 & 0 & -2 \end{bmatrix} \mathbf{v}(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \bar{e}_1(t) \\ \bar{e}_2(t) \end{bmatrix}$$

$$\begin{bmatrix} y(t) \\ e_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \mathbf{v}(t)$$

In these equations, $\bar{e}_1(t)$ is the zero-order-hold output with $e_1^*(t)$ as its input, and $\bar{e}_2(t)$ is defined in the same manner. To obtain the discrete state matrices, we use the

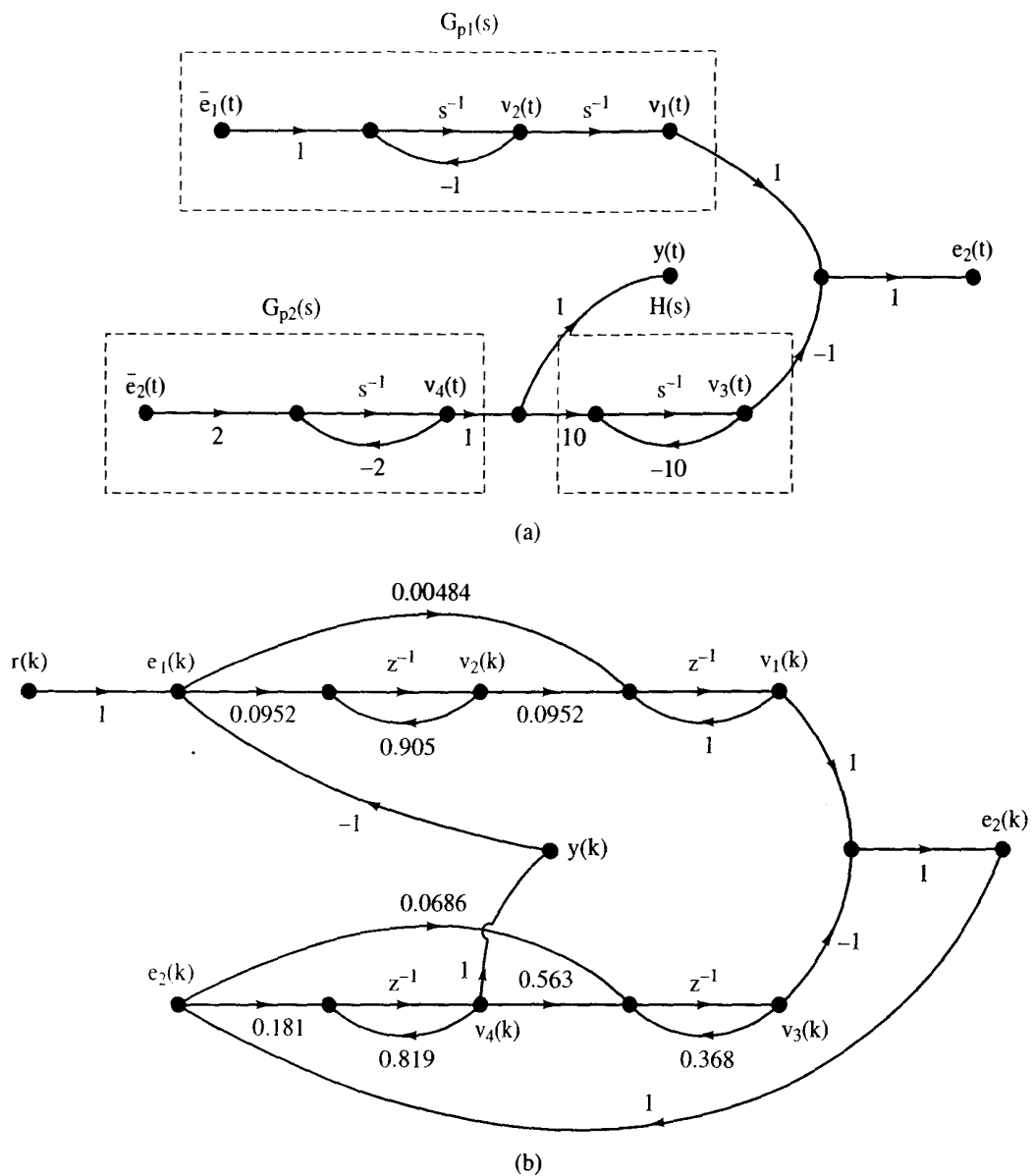


Figure 5-19 System for Example 5.5.

technique of Section 4.10 and, for this example, calculate the following matrices by computer.

$$\mathbf{v}(k+1) = \begin{bmatrix} 1 & 0.0952 & 0 & 0 \\ 0 & 0.905 & 0 & 0 \\ 0 & 0 & 0.368 & 0.563 \\ 0 & 0 & 0 & 0.819 \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} 0.00484 & 0 \\ 0.0952 & 0 \\ 0 & 0.0686 \\ 0 & 0.181 \end{bmatrix} \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix}$$

$$\begin{bmatrix} y(k) \\ e_2(k) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \mathbf{v}(k)$$

A simulation diagram of this system, with all external connecting paths, is shown in

Figure 5-19b. From this simulation diagram we write the discrete state equation for the closed-loop system.

$$\mathbf{v}(k+1) = \begin{bmatrix} 1 & 0.0952 & 0 & -0.00484 \\ 0 & 0.905 & 0 & -0.0952 \\ 0.0686 & 0 & 0.299 & 0.563 \\ 0.181 & 0 & -0.181 & 0.819 \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \\ 0 \\ 0 \end{bmatrix} r(k)$$

$$y(k) = [0 \ 0 \ 0 \ 1] \mathbf{v}(k)$$

Note the similarity of this technique to that developed earlier for finding closed-loop transfer functions. The system is opened at each sampler, each zero-order-hold output is assumed to be an input, and each sampler input is assumed to be an output. Discrete state equations are then written relating these specified inputs and outputs. These state equations are manipulated, through the use of a simulation diagram, to obtain the state equations of the closed-loop system.

The technique above can be applied to low-order systems; however, writing system equations from complex flow graphs is at best tenuous. Instead, a matrix procedure that can be implemented by a computer program is needed, and one will now be developed. Consider Example 5.5. The discrete state equations for the continuous system can be written as

$$\mathbf{v}(k+1) = \mathbf{A}_1 \mathbf{v}(k) + \mathbf{B}_1 \mathbf{e}(k) \quad (5-29)$$

$$\mathbf{e}(k) = \mathbf{C}_1 \mathbf{v}(k) + \mathbf{D}_1 r(k) \quad (5-30)$$

where $\mathbf{e}(t) = [e_1(t) \ e_2(t)]^T$. Thus these equations can be combined to yield

$$\mathbf{v}(k+1) = [\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_1] \mathbf{v}(k) + \mathbf{B}_1 \mathbf{D}_1 r(k) \quad (5-31)$$

which is the required equation. An example will now be given.

Example 5.6

Consider the system of Example 5.5. Since $e_1(k) = r(k) - y(k) = r(k) - v_4(k)$ and $e_2(k) = v_1(k) - v_3(k)$, the equations for $\mathbf{e}(k)$ can be written as

$$\begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r(k) = \mathbf{C}_1 \mathbf{v}(k) + \mathbf{D}_1 r(k)$$

Then, in (5-31),

$$\begin{aligned} \mathbf{B}_1 \mathbf{C}_1 &= \begin{bmatrix} 0.00484 & 0 \\ 0.0952 & 0 \\ 0 & 0.0686 \\ 0 & 0.181 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & -0.00484 \\ 0 & 0 & 0 & -0.0952 \\ 0.0686 & 0 & -0.0686 & 0 \\ 0.181 & 0 & -0.181 & 0 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_1 &= \begin{bmatrix} 1 & 0.0952 & 0 & 0 \\ 0 & 0.905 & 0 & 0 \\ 0 & 0 & 0.368 & 0.563 \\ 0 & 0 & 0 & 0.819 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & -0.00484 \\ 0 & 0 & 0 & -0.0952 \\ 0.0686 & 0 & -0.0686 & 0 \\ 0.181 & 0 & -0.181 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0.0952 & 0 & -0.00484 \\ 0 & 0.905 & 0 & -0.0952 \\ 0.0686 & 0 & 0.2994 & 0.563 \\ 0.181 & 0 & -0.181 & 0.819 \end{bmatrix} \end{aligned}$$

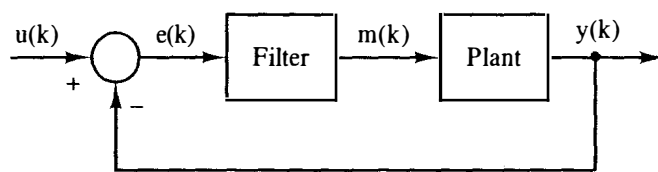
Also, in (5-31),

$$\mathbf{B}_1 \mathbf{D}_1 = \begin{bmatrix} 0.00484 & 0 \\ 0.0952 & 0 \\ 0 & 0.0686 \\ 0 & 0.181 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.00484 \\ 0.0952 \\ 0 \\ 0 \end{bmatrix}$$

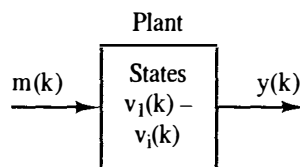
These results agree with those obtained in Example 5.5 and are less prone to error. Also, the matrix procedure of this example can be implemented on a digital computer.

As a final point, note that each system input must be sampled prior to being applied to an analog part of the system. If this is not the case, (5-30) cannot be written as a function of only $r(k)$, and no discrete model is possible. If an input that is not sampled varies slowly over a sample period (only low frequencies appear in the input signal), this input is often assumed to be sampled, even though the resultant discrete model is somewhat in error.

The system state equations are more difficult to derive if the system contains a digital controller. A single-loop digital control system is shown in Figure 5-20a. To



(a)



Filter

(b)

Figure 5-20 Discrete control system.

obtain the state equations, we consider the digital filter and the plant separately and write the state equations for these two parts. We assign states $v_1(k)$ through $v_i(k)$ to the plant, where i is the order of the plant. Then we assign states $v_{i+1}(k)$ through $v_n(k)$ to the filter, where $n - i$ is the order of the filter. Hence we can write the state equations

$$\mathbf{v}(k+1) = \mathbf{A}_1 \mathbf{v}(k) + \mathbf{B}_1 m(k) + \mathbf{B}_2 e(k) \quad (5-32)$$

since both $m(k)$ and $e(k)$ are inputs, as shown in Figure 5-20b. Next we write for the filter

$$m(k) = \mathbf{C}_1 \mathbf{v}(k) + D_1 e(k) \quad (5-33)$$

and for the plant

$$y(k) = \mathbf{C} \mathbf{v}(k) \quad (5-34)$$

Thus, for the feedback path and from (5-34), we write

$$e(k) = u(k) - y(k) = u(k) - \mathbf{C} \mathbf{v}(k) \quad (5-35)$$

We obtain the system state equations by eliminating $m(k)$ and $e(k)$ from (5-32), (5-33), and (5-35). From (5-33) and (5-35),

$$m(k) = \mathbf{C}_1 \mathbf{v}(k) + D_1 [u(k) - \mathbf{C} \mathbf{v}(k)] = [\mathbf{C}_1 - D_1 \mathbf{C}] \mathbf{v}(k) + D_1 u(k) \quad (5-36)$$

Then, substituting (5-35) and (5-36) into (5-32), we obtain

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbf{A}_1 \mathbf{v}(k) + \mathbf{B}_1 [(\mathbf{C}_1 - D_1 \mathbf{C}) \mathbf{v}(k) + D_1 u(k)] \\ &\quad + \mathbf{B}_2 [u(k) - \mathbf{C} \mathbf{v}(k)] \\ &= [\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_1 + (-\mathbf{B}_2 - D_1 \mathbf{B}_1) \mathbf{C}] \mathbf{v}(k) + [D_1 \mathbf{B}_1 + \mathbf{B}_2] u(k) \end{aligned} \quad (5-37)$$

which is the desired relationship. An example will now be given.

Example 5.7

The state equations for the system of Example 4.13 (shown in Figure 5-21a) will be developed. From Example 4.13, the state equations for the plant are

$$\begin{aligned} \begin{bmatrix} v_1(k+1) \\ v_2(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} + \begin{bmatrix} 0.0484 \\ 0.952 \end{bmatrix} m(k) \\ y(k) &= [1 \quad 0] \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \end{aligned}$$

The filter is modeled as shown in Figure 5-21b. The state equations for the filter are

$$\begin{aligned} v_3(k+1) &= 0.9v_3(k) + e(k) \\ m(k) &= (0.81 - 0.8)v_3(k) + 0.9e(k) \\ &= 0.01v_3(k) + 0.9e(k) \end{aligned}$$

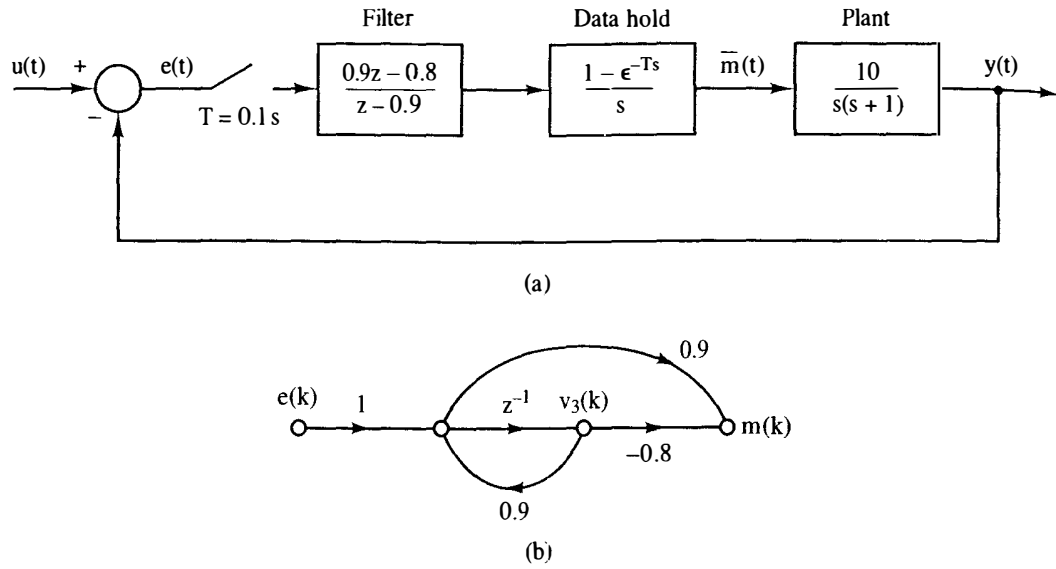


Figure 5-21 System for Example 5.7.

Combining the state equations for $\mathbf{v}(k)$, we obtain

$$\begin{bmatrix} v_1(k+1) \\ v_2(k+1) \\ v_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.0952 & 0 \\ 0 & 0.905 & 0 \\ 0 & 0 & 0.9 \end{bmatrix} \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} + \begin{bmatrix} 0.0484 \\ 0.952 \\ 0 \end{bmatrix} m(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} e(k)$$

Also, from Figure 5-21, since $e(t) = u(t) - y(t)$,

$$e(k) = u(k) - y(k) = u(k) - [1 \ 0 \ 0] \mathbf{v}(k)$$

and from above,

$$m(k) = [0 \ 0 \ 0.01] \mathbf{v}(k) + 0.9e(k)$$

Comparing the equation above for $\mathbf{v}(k)$ with (5-32), we see that

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0.0952 & 0 \\ 0 & 0.905 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 0.0484 \\ 0.952 \\ 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

From the equation above for $m(k)$ and (5-33),

$$\mathbf{C}_1 = [0 \ 0 \ 0.01], \quad D_1 = 0.9$$

and from the equation above for $e(k)$ and (5-35),

$$\mathbf{C} = [1 \ 0 \ 0]$$

Then, in (5-37),

$$\begin{aligned} \mathbf{B}_1 \mathbf{C}_1 &= \begin{bmatrix} 0.0484 \\ 0.952 \\ 0 \end{bmatrix} [0 \ 0 \ 0.01] = \begin{bmatrix} 0 & 0 & 0.000484 \\ 0 & 0 & 0.00952 \\ 0 & 0 & 0 \end{bmatrix} \\ (-\mathbf{B}_2 - D_1 \mathbf{B}_1) \mathbf{C} &= \begin{bmatrix} 0 - 0.04356 \\ 0 - 0.8568 \\ -1 - 0 \end{bmatrix} [1 \ 0 \ 0] \\ &= \begin{bmatrix} -0.04356 & 0 & 0 \\ -0.8568 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \end{aligned}$$

Thus

$$\begin{aligned} \mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_1 + (-\mathbf{B}_2 - D_1 \mathbf{B}_1) \mathbf{C} &= \begin{bmatrix} 0.9564 & 0.0952 & 0.000484 \\ -0.8568 & 0.905 & 0.00952 \\ -1 & 0 & 0.9 \end{bmatrix} \\ D_1 \mathbf{B}_1 + \mathbf{B}_2 &= \begin{bmatrix} 0.04356 + 0 \\ 0.8568 + 0 \\ 0 + 1 \end{bmatrix} \end{aligned}$$

Thus the state equations for this system are, from (5-37) and (5-34),

$$\begin{aligned} \mathbf{v}(k+1) &= \begin{bmatrix} 0.9565 & 0.0952 & 0.000484 \\ -0.8568 & 0.905 & 0.00952 \\ -1 & 0 & 0.9 \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} 0.04356 \\ 0.8568 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [1 \ 0 \ 0] \mathbf{v}(k) \end{aligned}$$

All calculations in this example are implemented in the computer program CTRL, described in Appendix VI.

Examples 5.6 and 5.7 illustrate the derivation of discrete state models for digital control systems. Of course, some systems are more complex than these in the examples above. However, the technique used to derive the state equations of (5-37) may be employed for more complex systems. For example, if $y(k)$ in (5-34) is also a function of $m(k)$, the derivation is somewhat more complicated (see Problem 5-26).

5.5 SUMMARY

system. From this specification, a sampled signal flow graph is derived which can be used to determine the Laplace transform and the z-transform of the output of the closed-loop system. In addition, a technique is developed for determining the

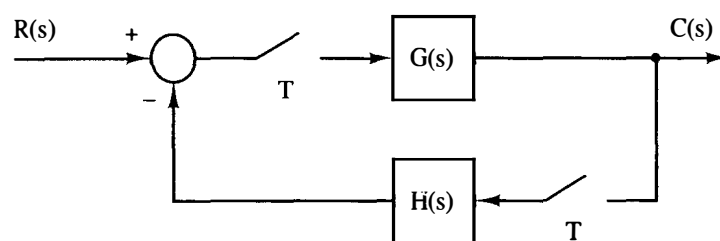
state-variable model of a closed-loop discrete-time system provided all inputs are sampled. In the following chapters these techniques will be utilized in analyzing and designing closed-loop discrete-time systems.

REFERENCES AND FURTHER READING

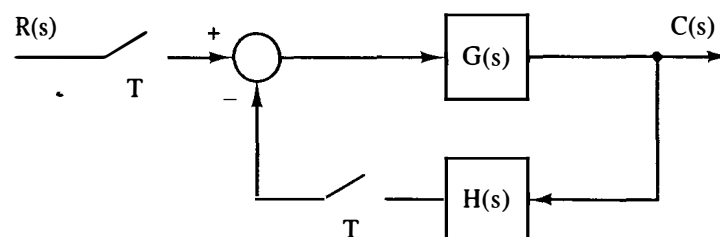
1. B. C. Kuo, *Digital Control Systems*, 2d ed. New York: Saunders College Publishing, 1992.
2. C. L. Phillips and S. M. Seltzer, "Design of Advanced Sampled-Data Control Systems," Contract DAAHO1-72-C-0901, Auburn University, Auburn, AL, July 1973.
3. J. A. Cadzow and H. R. Martens, *Digital-Time and Computer Control Systems*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1970.
4. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1988.
5. E. I. Jury, *Theory and Application of the z-Transform Method*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.

PROBLEMS

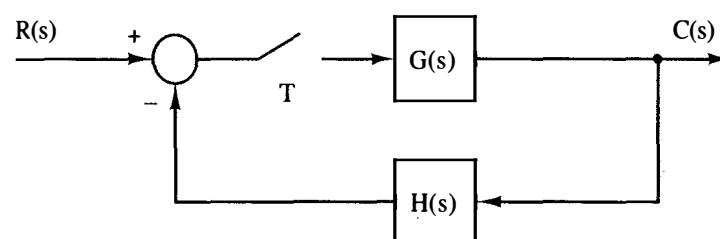
- 5-1. For each of the systems of Figure P5-1, express $C(z)$ as a function of the input and the transfer functions shown.
- 5-2. For each of the systems of Figure P5-2, express $C(z)$ as a function of the input and the transfer functions shown.
- 5-3. (a) Derive the transfer function $C(z)/R(z)$ for the system of Figure P5-1b.
 (b) Derive the transfer function $C(z)/R(z)$ for the system of Figure P5-1c.
 (c) Even though the two systems are different, the transfer functions are equal. Explain why this is true. *Hint*: Consider the error signal.
- 5-4. Consider the system of Figure P5-4.
 (a) Calculate the system output $C(z)$ for the signal from $G_2(s)$ disconnected from the middle summing junction.
 (b) Calculate the system output $C(z)$ for the signal from $G_2(s)$ disconnected from the last summing junction.
 (c) Calculate the system output $C(z)$ for the system as shown.
- 5-5. For the system of Figure P5-1a, suppose that the sampler in the forward path samples at $t = 0, T, 2T, \dots$, and the sampler in the feedback path samples at $t = T/2, 3T/2, 5T/2, \dots$.
 (a) Find the system transfer function $C(z)/R(z)$.
 (b) Suppose that both samplers operate at $t = T/2, 3T/2, 5T/2, \dots$. Find $C(z)$ in its simplest form.
- 5-6. The system of Figure P5-6 contains a digital filter with the transfer function $D(z)$. Express $\phi_m(z)$ as a function of the input. The roll-axis control system of the Pershing missile is of this configuration [2].
- 5-7. Consider the two-loop system of Figure P5-7. The gain K is used to give the inner loop certain specified characteristics. Then the controller $D(z)$ is designed to compensate



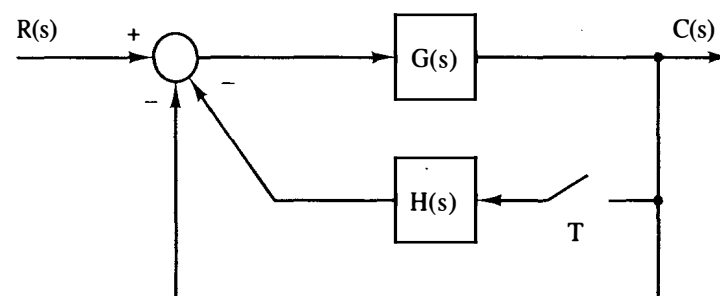
(a)



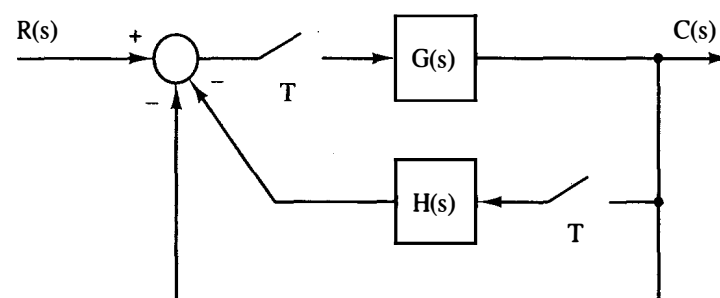
(b)



(c)



(d)



(e)

Figure P5-1 Systems for Problem 5-1.

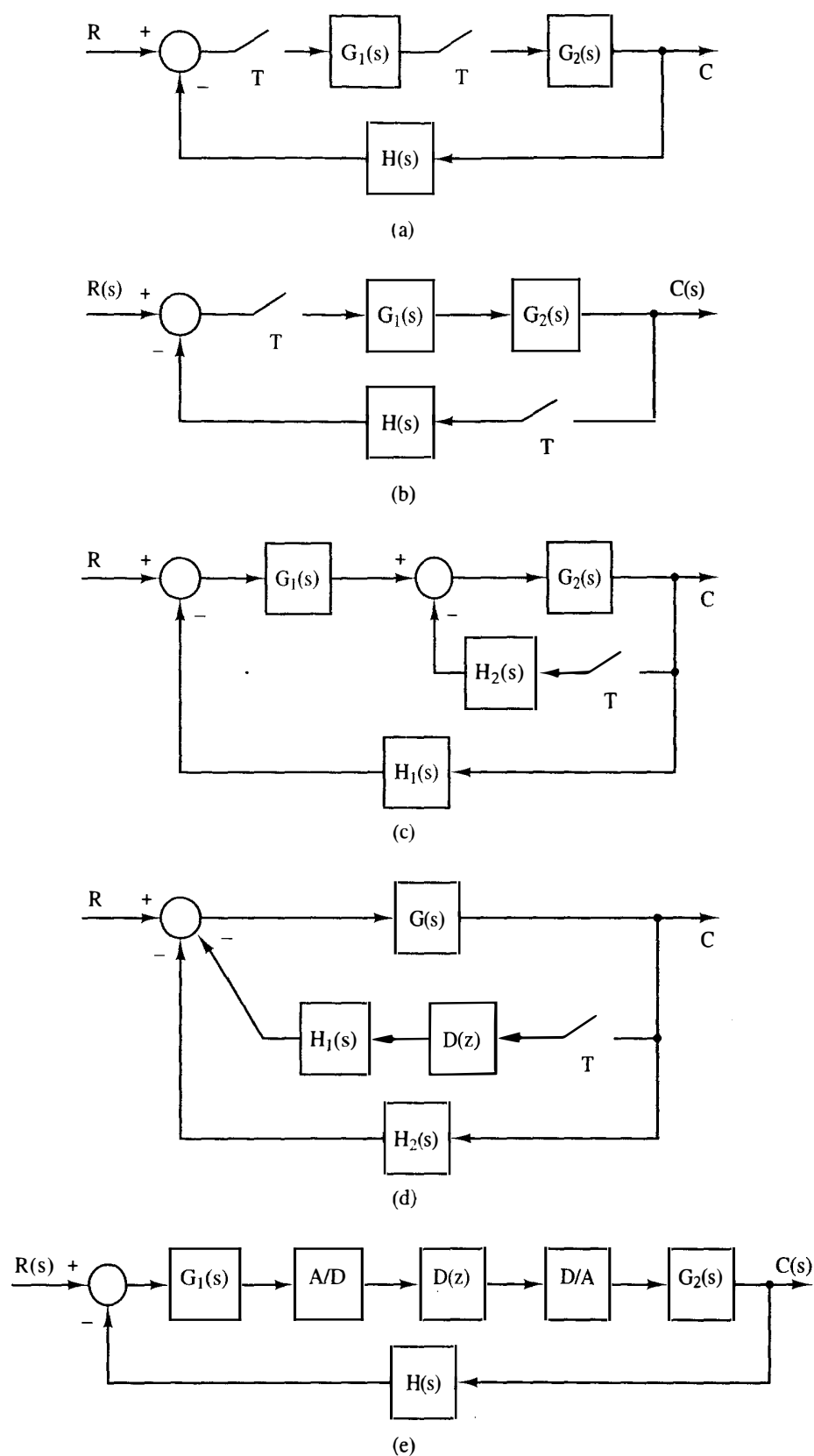


Figure P5-2 Systems for Problem 5-2.

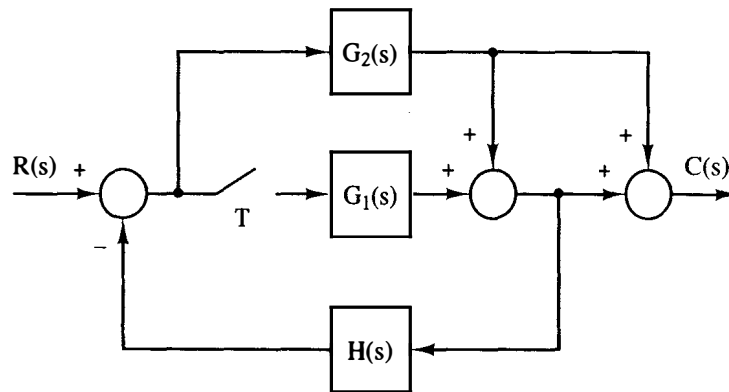


Figure P5-4 System for Problem 5-4.

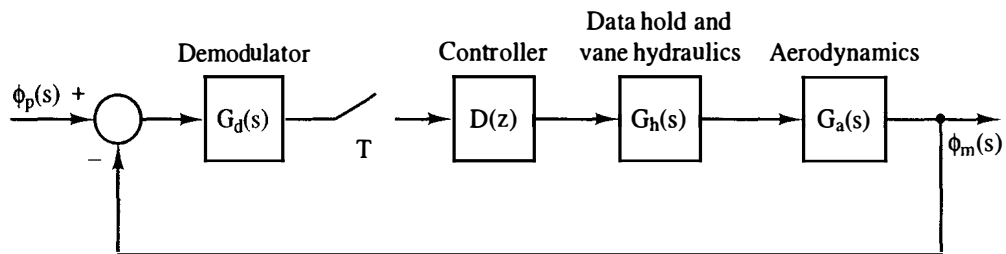


Figure P5-6 Roll-axis control system for a Pershing missile.

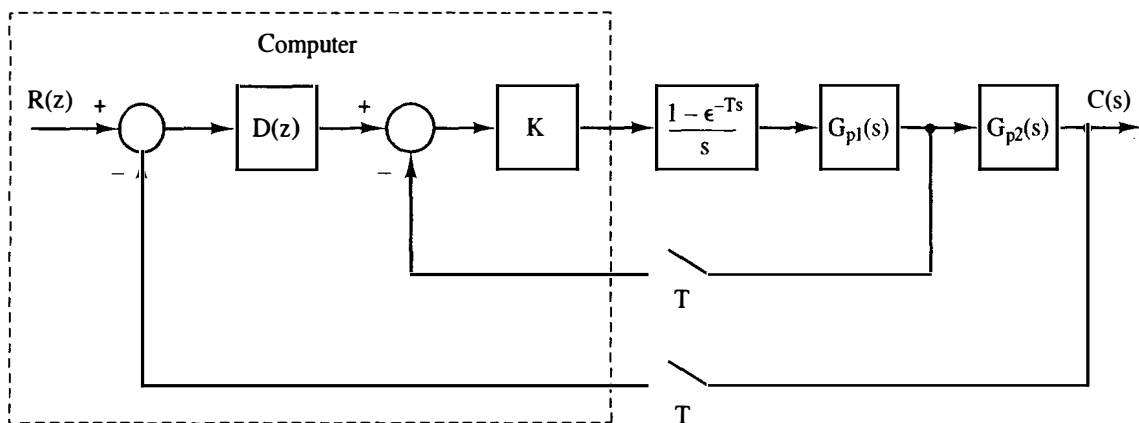


Figure P5-7 System for Problem 5-7.

the entire system. The input $R(z)$ is generated within the computer, and thus does not exist as a continuous signal. Solve for $C(z)$.

- 5-8. The system of Figure P5-8 is the same as that of Example 5.1, except that the sampler has been moved to the feedback path. This may occur for two reasons: (1) the sensor output is in sampled form, and (2) the input function is more conveniently generated in the computer.

(a) Calculate both $C(s)$ and $C(z)$ for the system of Figure P5-8. Note that $C(s)$ cannot contain the variable z .

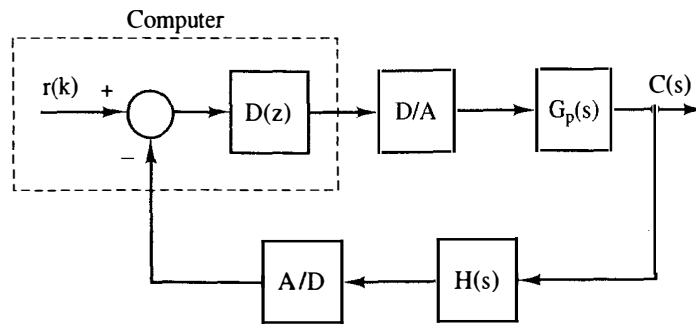


Figure P5-8 System for Problem 5-8.

- (b) Note that the output functions in part (a) are identical to those found in Example 5.1, even though the two configurations are different. Explain why.
- 5-9. For Figure P5-1, list, for each system, all transfer functions that contain the transfer function of a zero-order hold.
- 5-10. For Figure P5-2, list, for each system, all transfer functions that contain the transfer function of a zero-order hold.
- 5-11. In the system of Figure P5-11, the ideal time delay represents the time required to complete the computations in the computer.
- (a) Derive the output function $C(z)$ for this system.
- (b) Suppose that the ideal delay is associated with the sensor rather than the computer, and the positions of $H_2(s)$ and the ideal delay are reversed. Find $C(z)$ for this case.

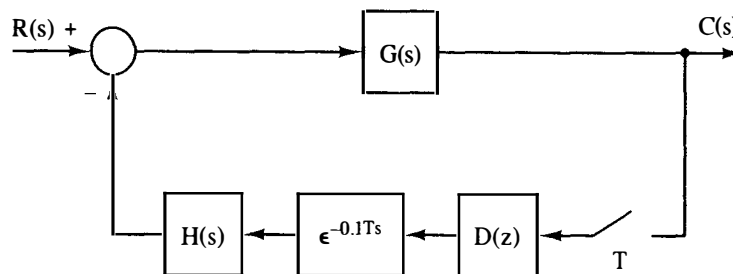


Figure P5-11 System for Problem 5-11.

- 5-12. Shown in Figure P5-12 is the block diagram for the temperature control system for a large test chamber. This system is described in Problem 1-10. The disturbance shown is the model of the effects of opening the chamber door. The following transfer functions are defined.

$$G(s) = \frac{2(1 - e^{-Ts})}{s(s + 0.5)}, \quad G_d(s) = \frac{2.5}{s + 0.5}, \quad H_k = 0.04$$

- (a) Derive the transfer function $C(z)/R(z)$, in terms of the transfer functions just defined.
- (b) With $r(t) = 0$, solve for the output function $C(s)$ in terms of the disturbance input and the transfer functions just defined.
- (c) Use superposition and the results of parts (a) and (b) to write the complete expression of $C(z)$.

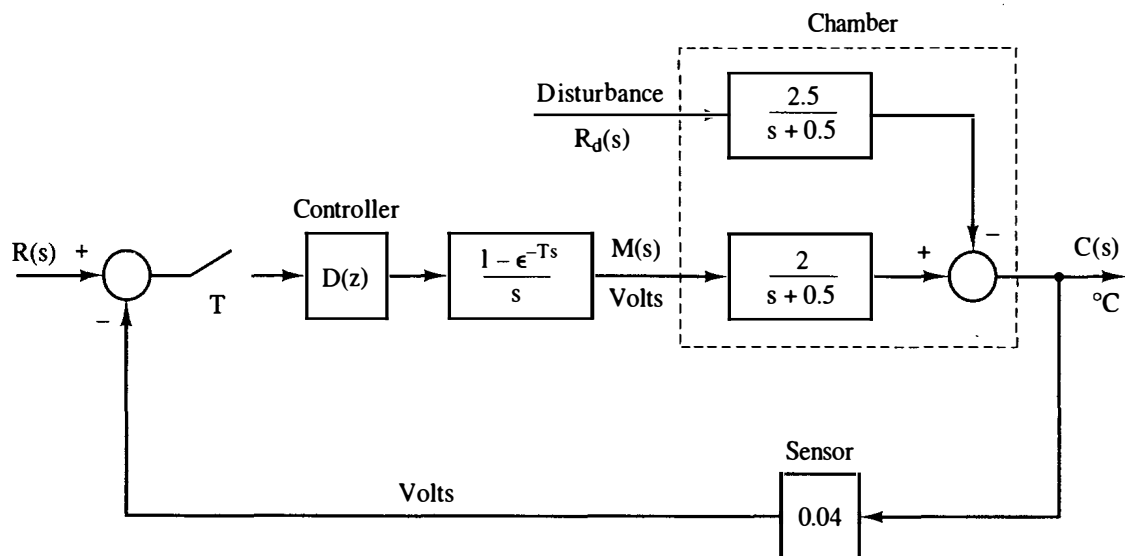


Figure P5-12 Chamber temperature control system.

5-13. Consider the robot-joint control system of Figure P5-13. This system is described in Problem 1-16.

- The sensor input is θ_a in degrees and the output is in volts. If the robot joint movement is mechanically restricted to $\pm 135^{\circ}$, find the range of the sensor output voltage. What should be the input voltage range for the A/D?
- Let $G_p(s)$ be the transfer function of the servomotor and gears, and $H_k = 0.07$ be the sensor gain. Find the system transfer function as a function of K , $G_p(s)$, and so on.
- Evaluate the system transfer function for $K = 2.4$, $T = 0.1$ s, and $D(z) = 1$.
- Verify the results in part (c) using MATLAB.

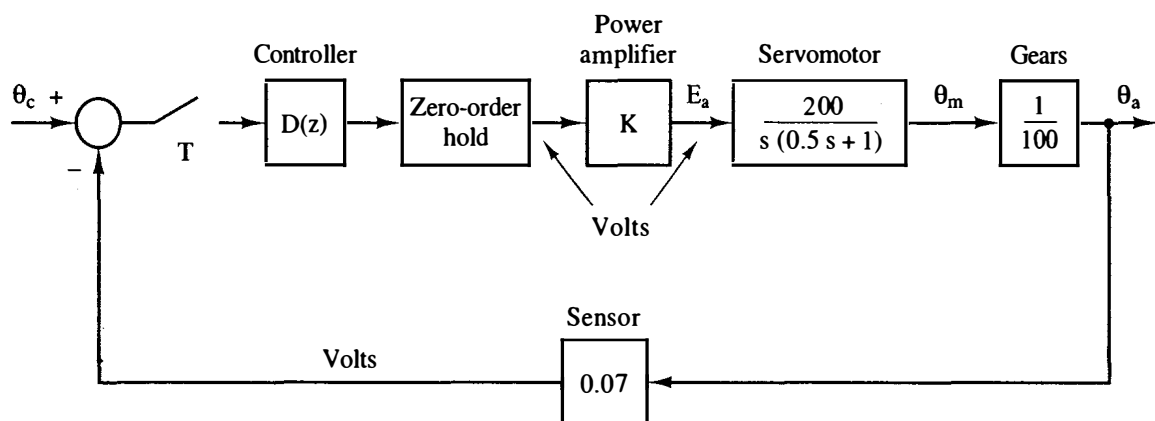


Figure P5-13 Robot arm joint control system.

5-14. Consider the satellite control system of Figure P5-14. The units of the attitude angle $\theta(t)$ is degrees, and the range is 0 to 360° . The sensor gain is $H_k = 0.02$.

- Suppose that the input ranges for available A/Ds are 0 to 5 V, 0 to 10 V, 0 to 20 V, ± 5 V, ± 10 V, and ± 20 V. Which range should be chosen? Why?
- The input signal $r(t)$ is a voltage. Find the required value of $r(t)$ to command the satellite attitude angle $\theta(t)$ to be 70° .

- (c) Repeat parts (a) and (b) if the range of $\theta(t)$ is $\pm 180^\circ$.
 (d) Let $G_p(s) = 1/(Js^2)$, the satellite transfer function. Find the system transfer function as a function of the transfer functions $D(z)$, K , and so on.
 (e) Evaluate the system transfer function for $D(z) = 1$, $T = 1$ s, $K = 2$, and $J = 0.1$.
 (f) Verify the results in part (e) using MATLAB.

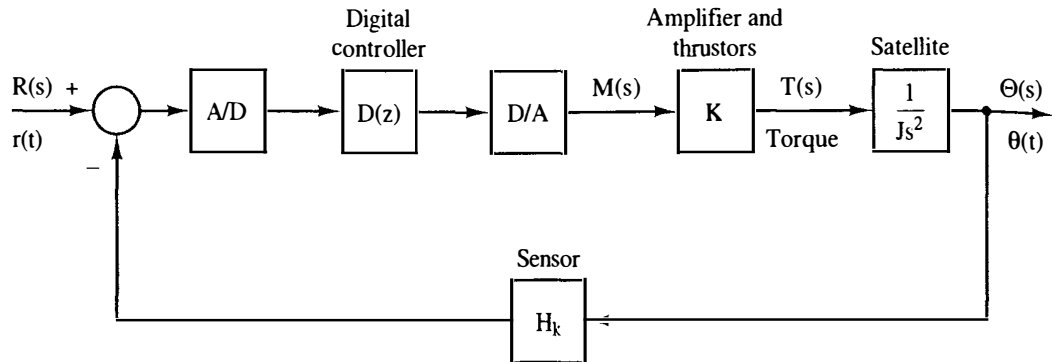


Figure P5-14 Block diagram for a satellite control system.

- 5-15. Consider the antenna control system of Figure P5-15. The units of the antenna angle $\theta(t)$ is degrees, and the range is $\pm 45^\circ$.
 (a) The input signal $r(k)$ is generated in the computer. Find the required values of $r(k)$ to command the satellite attitude angle $\theta(t)$ to be 30° and to be -30° .
 (b) Let $G_p(s) = 20/(s^2 + 6s)$, the transfer function of the motor, gears and pedestal. Find the system transfer function as a function of the transfer functions $D(z)$, K , and so on.
 (c) Evaluate the system transfer function for $D(z) = 1$, $T = 0.05$ s, and $K = 20$.
 (d) Verify the results in part (c) using MATLAB.

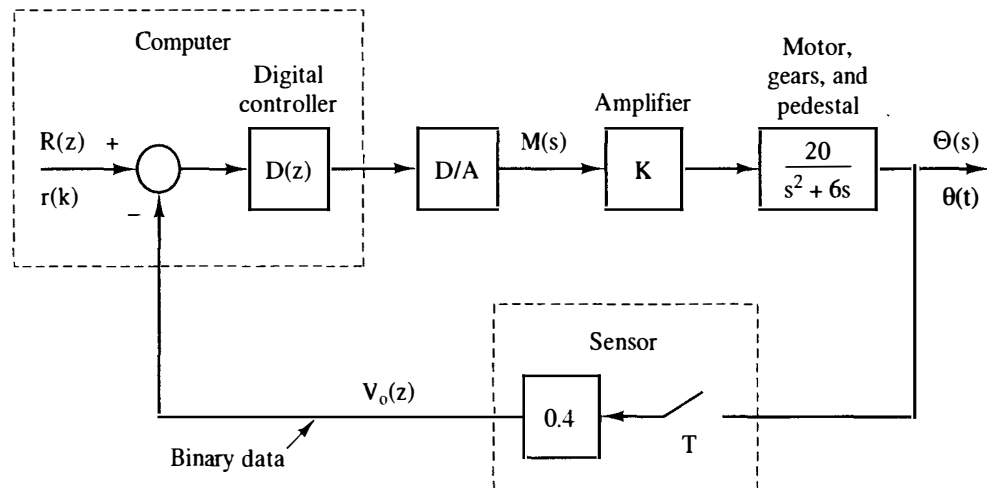


Figure P5-15 Block diagram for an antenna control system.

- 5-16. Given a closed-loop system described by the transfer function

$$\frac{C(z)}{R(z)} = \frac{z^2 + 0.3z + 0.2}{z^2 - z + 0.9}$$

- (a) Express $c(k)$ as a function of $r(k)$, as a single difference equation.
 (b) Find a set of state equations for this system.
 (c) Calculate the transfer function from the results of part (b), to verify these results.
 (d) Verify the results in part (c) using MATLAB.

5-17. Repeat Problem 5-16 for each of the transfer functions

(a) $\frac{C(z)}{R(z)} = \frac{0.1z}{z - 0.9}$

(b) $\frac{C(z)}{R(z)} = \frac{0.1}{z - 0.9}$

(c) $\frac{C(z)}{R(z)} = \frac{0.2z - 0.05}{z - 0.9}$

(d) $\frac{C(z)}{R(z)} = \frac{0.8z + 0.7}{z^2 - 1.6z + 0.8}$

5-18. For the system of Figure P5-1(a), let $T = 0.1$ s and

$$G(s) = \frac{1 - e^{-Ts}}{s^2}, \quad H(s) = \frac{1 - e^{-Ts}}{s^2 + s}$$

- (a) Calculate $G(z)$ and $H(z)$.
 (b) Draw simulation diagrams for $G(z)$ and $H(z)$, and interconnect these diagrams to form the control system of Figure P5-1a.
 (c) Write the discrete state equations for part (b).
 (d) Find the system characteristic equation, using the transfer functions of part (a).
 (e) Show that the state model in part (c) has the same characteristic equation as in part (d).
- 5-19. Let $T = 0.1$ s for the system of Figure P5-19. Derive a set of discrete state equations for the closed-loop system, for the plant described by each of the differential equations.

(a) $\frac{dy(t)}{dt} + 2y(t) = 3\bar{m}(t)$

Choose the state variable to be $y(kT)$.

(b) $\frac{d^2y(t)}{dt^2} + 3\frac{dy(t)}{dt} + 2y(t) = 6\bar{m}(t)$

Choose the state variables to be $y(kT)$ and $dy(t)/dt|_{t=kT}$.

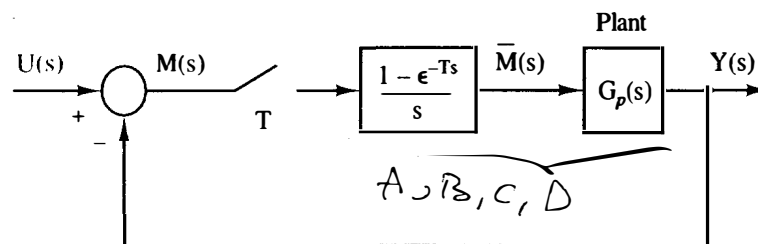


Figure P5-19 System for Problem 5-19.

5-20. Suppose that the plant in Figure P5-19 has the discrete state model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}m(k)$$

$$y(k) = \mathbf{C}\mathbf{x}(k) + Dm(k)$$

Derive the state model for the closed-loop system, in terms of \mathbf{A} , \mathbf{B} , \mathbf{C} , and D .

- 5-21.** Find a discrete state variable model of the closed-loop system shown in Figure P5-19 if the discrete state model of the plant is given by:

(a) $x(k+1) = 0.7x(k) + 0.3m(k)$

$$y(k) = 0.2x(k) + 0.5m(k)$$

(b) $x(k+1) = \begin{bmatrix} 0 & 1 \\ -0.9 & 1.3 \end{bmatrix} x(k) + \begin{bmatrix} 0.1 \\ 0.05 \end{bmatrix} m(k)$

$$y(k) = [1.2 \quad -0.7] x(k)$$

(c) $x(k+1) = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.9 & 1 \\ -1 & 0 & 0.9 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} m(k)$

$$y(k) = [1 \quad 0 \quad 0] x(k)$$

- 5-22.** Consider the temperature control system of Problem 5-12 and Figure P5-12. Suppose that the digital filter transfer function is given by

$$D(z) = 1.1 + \frac{0.02z}{z-1}$$

which is a PI (proportional-integral) controller. Suppose that $T = 0.6$ s, and let $R_d(s) = 0$ (ignore the disturbance input).

- Using the closed-loop transfer function, derive a discrete state model for the system.
 - Derive a discrete state model for the plant from the plant transfer function. Then derive the state model of the closed-loop system by adding the filter and the feedback path to the flow graph of the plant.
 - Calculate the transfer function from the state model of part (b), to verify these results.
 - How are the states of parts (a) and (b) related? Do not solve for the exact relationship.
- 5-23.** Consider the satellite control system of Problem 5-14 and Figure P5-14. Let $D(z) = 1$, $T = 1$ s, $K = 2$, $J = 0.1$, and $H_k = 0.02$.
- Using the closed-loop transfer function, derive a discrete state model for the system.
 - Derive a discrete state model for the plant from the plant transfer function. Then derive a state model for the closed-loop system by adding the feedback path and system input to a flow graph of the plant.
 - Calculate the system transfer function from the state model found in part (b), to verify the state model.
 - Verify the results in part (c) using MATLAB.
- 5-24.** Consider the antenna control system of Problem 5-15 and Figure P5-15. Let $D(z) = 1$, $T = 0.05$ s, and $K = 20$.
- Using the closed-loop transfer function, derive a discrete state model for the system in the observer canonical form of Figure 2-10.
 - Derive a discrete state model for the plant from the plant transfer function. Then derive a state model for the closed-loop system by adding the feedback path and system input to a flow graph of the plant.
 - Calculate the system transfer function from the state model found in part (b), to verify the state model.

(d) Verify the results in part (c) using MATLAB.

5-25. Consider the robot joint control system of Problem 5-13 and Figure P5-13. Let $D(z) = 1$, $T = 0.1$ s, and $K = 2.4$.

- (a) Using the closed-loop transfer function, derive a discrete state model for the system.
- (b) Derive a discrete state model for the plant from the plant transfer function. Then derive a state model for the closed-loop system by adding the feedback path and system input to a flow graph of the plant.
- (c) Calculate the system transfer function from the state model found in part (b), to verify the state model.

5-26. Suppose that, for the system of Figure 5-20, equation (5-34) is

$$y(k) = C\mathbf{v}(k) + d_2m(k)$$

- (a) Derive the state model of (5-37) for this case.
- (b) This system has an algebraic loop. Identify this loop.
- (c) The gain of the algebraic loop is $-d_1d_2$. What is the effect on the system equations if $d_1d_2 = -1$?
- (d) We can argue that algebraic loops as in this case cannot occur in physical systems, since time delay is always present in signal transmission. Quite often we can ignore this delay. Under what conditions can we obviously *not* ignore delay in this system?

System Time-Response Characteristics

6.1 INTRODUCTION

In this chapter we consider five important topics. First, the time response of a discrete-time system is investigated. Next, regions in the s -plane are mapped into regions in the z -plane. Then by using the correlation between regions in the two planes, the effect of the closed-loop z -plane poles on the system transient response is discussed. Next, the effects of the system transfer characteristics on the steady-state system error are considered. Finally, the simulation of analog and discrete-time systems is introduced.

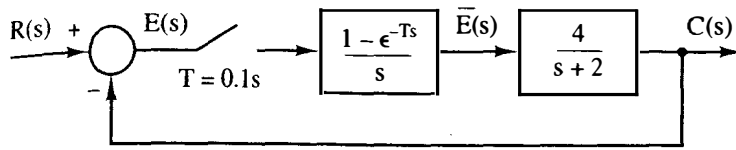
6.2 SYSTEM TIME RESPONSE

In this section the time response of discrete-time systems is introduced via examples. In these examples some of the techniques of determining the system time response are illustrated.

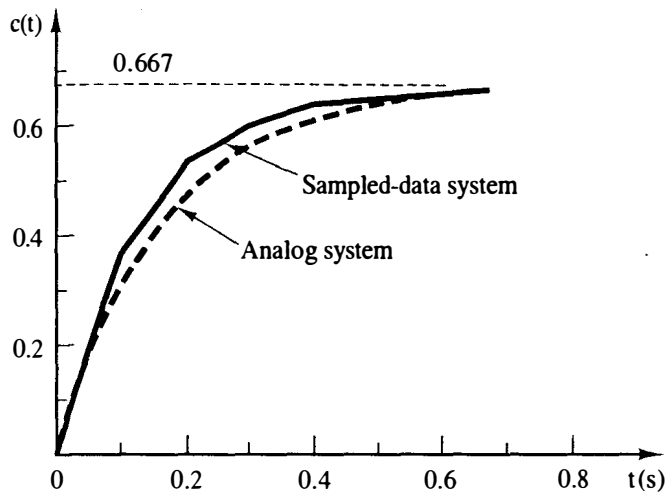
Example 6.1

The unit-step response will be found for the first-order system in Figure 6-1a. Since the plant of a temperature control system is often modeled as a first-order system, this system might then be the model of a temperature control system (see Section 1.6). Using the techniques developed in Chapter 5, we can express the system output as

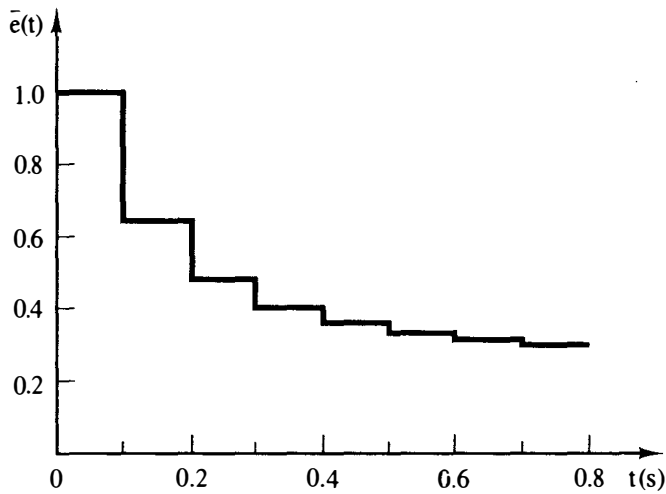
$$C(z) = \frac{G(z)}{1 + G(z)} R(z)$$



(a)



(b)



(c)

Figure 6-1 System and response for Example 6.1.

where $G(z)$ is defined as

$$\begin{aligned} G(z) &= \mathcal{Z} \left[\frac{1 - e^{-Ts}}{s} \frac{4}{s + 2} \right] = \frac{z - 1}{z} \mathcal{Z} \left[\frac{4}{s(s + 2)} \right] \\ &= \frac{z - 1}{z} \frac{2(1 - e^{-2T})z}{(z - 1)(z - e^{-2T})} = \frac{0.3625}{z - 0.8187}, \quad T = 0.1 \text{ s} \end{aligned}$$

from the transform table in Appendix VIII. Thus the closed-loop transfer function $T(z)$ is given by

$$T(z) = \frac{G(z)}{1 + G(z)} = \frac{0.3625}{z - 0.4562}$$

Since $R(z) = \mathcal{Z}[1/s] = z/(z - 1)$,

$$C(z) = \frac{0.3625z}{(z - 1)(z - 0.4562)} = \frac{0.667z}{z - 1} + \frac{-0.667z}{z - 0.4562}$$

The inverse z -transform of this function yields the system time response at the sampling instants. Thus

$$c(kT) = 0.667[1 - (0.4562)^k]$$

This response is listed in Table 6-1. It is seen that the response reaches a steady-state value of 0.667.

Example 6.2

This example is a continuation of Example 6.1. To show some of the effects of sampling on the system response, we will remove the sampler and zero-order hold, and solve for the unit-step response of the resulting analog system. The closed loop transfer function $T_a(s)$ is given by

$$T_a(s) = \frac{G_p(s)}{1 + G_p(s)} = \frac{4}{s + 6}$$

where $G_p(s) = 4/(s + 2)$ is the plant transfer function. Hence the analog system unit-step response is given by

$$C_a(s) = \frac{4}{s(s + 6)} = \frac{0.667}{s} + \frac{-0.667}{s + 6}$$

and

$$c_a(t) = 0.667(1 - e^{-6t})$$

This response is also listed in Table 6-1. Both step responses are plotted in Figure 6-1b. We may also calculate the response at all instants of time for the sampled-data

TABLE 6-1
RESPONSES FOR
EXAMPLE 6.1

kT	$c(kT)$	$c_a(t)$
0	0	0
0.1	0.363	0.300
0.2	0.528	0.466
0.3	0.603	0.557
0.4	0.639	0.606
0.5	0.654	0.634
0.6	0.661	0.648
\vdots		
1.0	0.666	0.665

system. The continuous output of the system of Figure 6-1a is given by

$$\begin{aligned} C(s) &= G(s) \left[\frac{R(z)}{1 + G(z)} \right]_{z = \epsilon^{Ts}} \\ &= \frac{4(1 - \epsilon^{-Ts})}{s(s+2)} \left[\frac{\frac{z}{z-1}}{1 + G(z)} \right]_{z = \epsilon^{Ts}} \\ &= \frac{4}{s(s+2)} \left[\frac{1}{1 + G(z)} \right]_{z = \epsilon^{Ts}} \end{aligned}$$

In this expression,

$$\begin{aligned} \frac{1}{1 + G(z)} &= \frac{1}{1 + \frac{0.3625}{z - 0.8187}} = \frac{z - 0.8187}{z - 0.4562} \\ &= 1 - 0.363z^{-1} - 0.165z^{-2} - \dots \end{aligned}$$

In $C(s)$ above, let the first factor be denoted as $C_1(s)$, that is,

$$C_1(s) = \frac{4}{s(s+2)} = \frac{2}{s} - \frac{2}{s+2}$$

Then

$$c_1(t) = 2(1 - \epsilon^{-2t})$$

Thus the output $C(s)$ is

$$C(s) = C_1(s)[1 - 0.363\epsilon^{-Ts} - 0.165\epsilon^{-2Ts} - \dots]$$

and hence

$$\begin{aligned} c(t) &= 2(1 - \epsilon^{-2t}) - 0.363(2)(1 - \epsilon^{-2(t-T)})u(t-T) \\ &\quad - 0.165(2)(1 - \epsilon^{-2(t-2T)})u(t-2T) - \dots \end{aligned}$$

For example, since $T = 0.1$ s,

$$\begin{aligned} c(3T) &= c(0.3) = 2(1 - \epsilon^{-0.6}) - 0.363(2)(1 - \epsilon^{-0.4}) \\ &\quad - 0.165(2)(1 - \epsilon^{-0.2}) = 0.603 \end{aligned}$$

This value checks that calculated by the z -transform approach and listed in Table 6-1. We see then the reason for the unusual shape of $c(t)$ in Figure 6-1b. This response is the superposition of a number of delayed step responses of the open-loop system. The steps appear as a result of the sampler and zero-order hold. For example, for $0 \leq t < 0.1$ s,

$$c(t) = 2(1 - \epsilon^{-2t})$$

Note that the time response of a sampled-data system of the configuration of that in Figure 6-1a is always the superposition of a number of step responses, independent of the form of the system input function. The steps in the input to the plant are also shown in the plot of the zero-order hold output, $\bar{e}(t)$, in Figure 6-1c.

Note the difficulty in calculating the continuous output $c(t)$ as compared to calculating the output $c(kT)$. For this reason, we seldom calculate the continuous output; if $c(t)$ is needed, it is obtained by simulation. In fact, in practical situations, we generally calculate all time responses by simulation. Simulation is introduced in Section 6.6.

Example 6.3

We will consider the system of Examples 6.1 and 6.2 further. Recall that the response of a first-order analog system has the transient-response term $ke^{-t/\tau}$, where τ is the *time constant*. We see then that the analog system of Figure 6-1a has a time constant of 0.167 s. A rule of thumb often used for selecting sample rates is that a rate of at least five samples per time constant is a good first choice. (Later results from experimentation with the system model and the actual physical system may indicate that a different rate is required.) For this system, we expect that reducing the sample period would decrease the effects of sampling, and that the sampled-data system characteristics would approach those of the analog system. In fact, if T is chosen to be 0.04 s, the unit-step response of the sampled-data system of Example 6.1 is essentially the same as that of the analog system. This point is illustrated further in the next example.

As an additional point, the final value of the unit-step response of the sampled-data system can be calculated using the final value theorem of the z -transform.

$$\begin{aligned}\lim_{n \rightarrow \infty} c(nT) &= (z - 1)C(z)|_{z=1} = (z - 1) \frac{G(z)}{1 + G(z)} R(z)|_{z=1} \\ &= (z - 1) \frac{G(z)}{1 + G(z)} \cdot \frac{z}{z - 1} \Big|_{z=1} = \frac{G(z)}{1 + G(z)} \Big|_{z=1} \\ &= \frac{G(1)}{1 + G(1)} = \frac{2}{1 + 2} = 0.667\end{aligned}$$

This value checks that in Table 6-1. Note that this derivation is general. Since the system input is a constant value of unity, we see from this derivation that the dc gain of a stable sampled-data system is simply the closed-loop transfer function evaluated at $z = 1$ (see Section 4.3).

A final point will be made concerning this example. For a continuous-time system whose output is

$$C(s) = G_p(s)E(s)$$

the dc gain is given by (see Section 4.3)

$$\text{dc gain} = \lim_{s \rightarrow 0} G_p(s)$$

and this value is $G_p(0)$ if $c(t)$ has a final value for a constant input. In the system of Figure 6-1a, if the input to the sampler is constant, the output of the zero-order hold is also constant and equal to the sampler input. Thus the sampler and data hold have no effect and may be removed, resulting in a continuous-time system. Thus, for a stable sampled-data system, the system dc gain may be found by removing the sampler and data hold, and evaluating the resulting system transfer function at $s = 0$. For this example, the open-loop dc gain is

$$G_p(s) \Big|_{s=0} = \frac{4}{s + 2} \Big|_{s=0} = 2$$

Thus the closed-loop dc gain is $2/(1 + 2) = 0.667$, which agrees with the dc gain calculated above via the z -transform.

Two important points were made in this example concerning the calculation of the steady-state gain of a sampled-data system with a constant input applied (i.e., the calculation of the dc gain). For a stable system with a constant input, the system output approaches a constant value as time increases for a constant input. The dc gain may be calculated by evaluating the transfer function with $z = 1$. In addition, the same value of dc gain is obtained by evaluating the transfer function of the analog system (sampler and zero-order hold removed) with $s = 0$. This second calculation applies in any case that the input to the sampler is constant.

Example 6.4



The system for this example is shown in Figure 6-2. As in the first example, we will calculate the unit-step response. This system will also appear in many of the following examples. As was demonstrated in Chapter 5, the system output can be expressed as

$$C(z) = \frac{G(z)}{1 + G(z)} R(z)$$

where, from the tables in Appendix VIII,

$$\begin{aligned} G(z) &= \left(\frac{z-1}{z} \right) \mathcal{Z} \left[\frac{1}{s^2(s+1)} \right]_{T=1} = \frac{z-1}{z} \left[\frac{z[(1-1+\epsilon^{-1})z + (1-\epsilon^{-1}-\epsilon^{-1})]}{(z-1)^2(z-\epsilon^{-1})} \right] \\ &= \frac{0.368z + 0.264}{z^2 - 1.368z + 0.368} \end{aligned}$$

Then

$$\frac{G(z)}{1 + G(z)} = \frac{0.368z + 0.264}{z^2 - z + 0.632}$$

Since

$$R(z) = \frac{z}{z-1}$$

then

$$\begin{aligned} C(z) &= \frac{z(0.368z + 0.264)}{(z-1)(z^2 - z + 0.632)} = 0.368z^{-1} + 1.00z^{-2} + 1.40z^{-3} \\ &\quad + 1.40z^{-4} + 1.15z^{-5} + 0.90z^{-6} + 0.80z^{-7} + 0.87z^{-8} \\ &\quad + 0.99z^{-9} + 1.08z^{-10} + 1.08z^{-11} + 1.00z^{-12} + 0.98z^{-13} \\ &\quad + \dots \end{aligned} \quad (6-1)$$

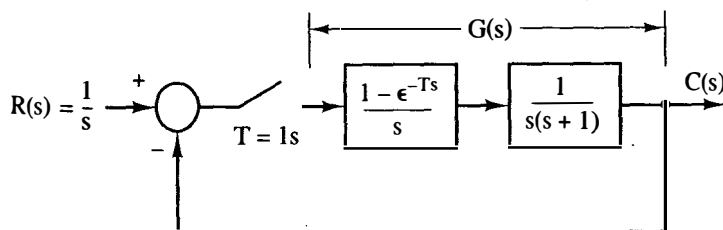


Figure 6-2 System used in Example 6.4.

The final value of $c(nT)$, obtained using the final-value theorem, is

$$\lim_{n \rightarrow \infty} c(nT) = \lim_{z \rightarrow 1} (z - 1)C(z) = \frac{0.632}{0.632} = 1$$

The step response for this system is plotted in Figure 6-3. The response between sampling instants was obtained from a simulation of the system, and, of course, the response at the sampling instants is given in (6-1). Also plotted in Figure 6-3 is the response of the system with the sampler and data hold removed. For this continuous-time system, the transfer function is, in standard notation [1],

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1}{s^2 + s + 1}$$

Therefore, $\omega_n = 1$, $\zeta = 0.5$, and hence the overshoot is approximately 18% (see Figure 6-4 below). Thus, as shown in Figure 6-3, the sampling has a destabilizing effect on the system. In general, it is desirable that the effects of sampling be negligible, that is, that the continuous system response and the discrete system response be approximately equal. For this system the sampling period and the plant time constant are equal. Hence the sampling frequency is too low, and should be increased, if allowed by hardware constraints. These effects will be discussed in detail later.

The system time response can also be calculated using a difference-equations approach. From the expression for the closed-loop transfer function

$$\frac{C(z)}{R(z)} = \frac{0.368z^{-1} + 0.264z^{-2}}{1 - z^{-1} + 0.632z^{-2}} \quad (6-2)$$

or

$$C(z)[1 - z^{-1} + 0.632z^{-2}] = R(z)[0.368z^{-1} + 0.264z^{-2}] \quad (6-3)$$

Taking the inverse z-transform of (6-3), we obtain the difference equation

$$\begin{aligned} c(kT) = & 0.368r[(k-1)T] + 0.264r[(k-2)T] \\ & + c[(k-1)T] - 0.632c[(k-2)T] \end{aligned} \quad (6-4)$$

Both $c(kT)$ and $r(kT)$ are zero for $k < 0$. Thus, from (6-4), $c(0) = 0$ and $c(1) = 0.368$.

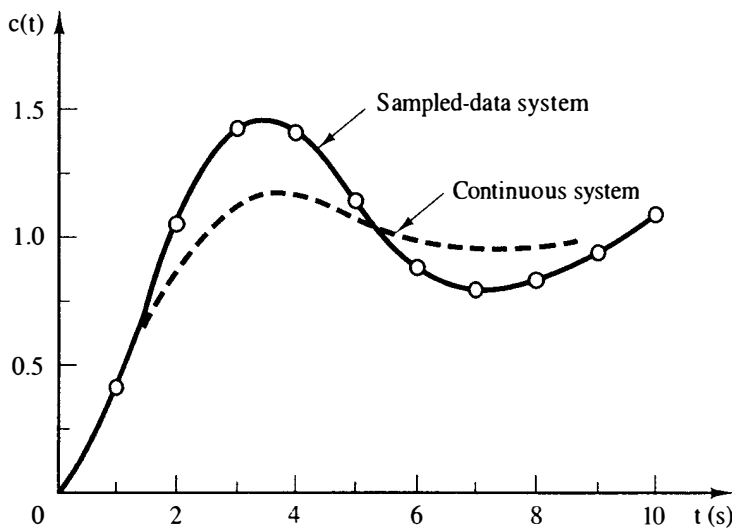
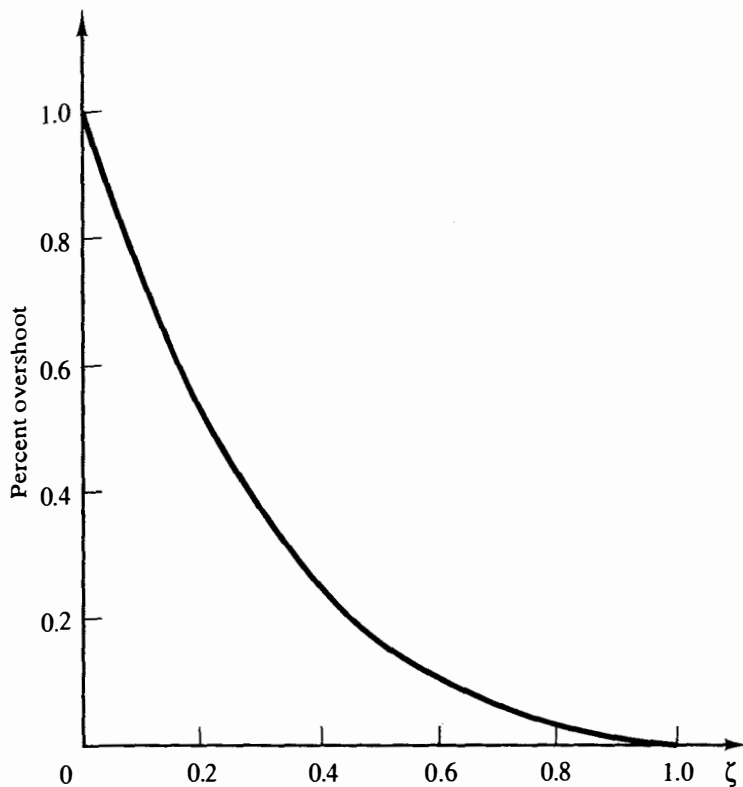


Figure 6-3 Step response of the systems analyzed in Example 6.4.

Figure 6-4 Percent overshoot versus ζ .

For $k \geq 2$, (6-4) becomes

$$c(kT) = 0.632 + c[(k-1)T] - 0.632c[(k-2)T]$$

Solving (6-4) for $c(kT)$ yields the same values as found in (6-1). A MATLAB program that solves (6-4) for the first 11 values of $c(kT)$ is given by

```
rm1 = 0; rm2 = 0; cm1 = 0; cm2 = 0;
for kk = 1:11
    k = kk - 1;
    r = 1;
    c = 0.368*rm1 + 0.264*rm2 + cm1 - 0.632*cm2;
    [k,c]
    cm2 = cm1; cm1 = c; rm2 = rm1; rm1 = r;
end
```

In this program, $r = r(k)$, $rm1 = r(k-1)$, $rm2 = r(k-2)$, and so on.

An additional point will be made concerning Example 6.4. The *damping ratio* ζ is an important indicator for complex poles in a transfer function. If the transfer function is second order of the standard form

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where ω_n is the *natural frequency*, the percent overshoot in the step response is given by [1].

$$\text{Percent overshoot} = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \times 100$$

This relationship is plotted in Figure 6-4. For systems that are not modeled as the standard second-order transfer function, Figure 6-4 gives an indication of the overshoot (or looseness) that will appear in a system response if the system transfer function has complex poles.

In this section the time responses of two different sampled-data systems were calculated using a z -transform approach. In addition, the continuous output was calculated for the first example, using the Laplace transform. Because of the complexities involved, it should be evident to the reader that in general we do not calculate the continuous output. In fact, system response is normally determined from either a digital simulation or a hybrid simulation of the system, and not from a transform approach. For high-order systems, simulation is the only practical technique for calculating the time response.

In many of the examples presented, the sampling frequency has been purposely chosen low, for two reasons. First, a sample period of 1 s is often selected to make the numerical calculations simpler. Second, if the sample period is large, only a few terms in the series expansion of $C(z)$ are required to give a good indication of the character of the response. In Example 6.3, if $T = 0.1$ s, the response of the sampled-data system is approximately the same as that of the continuous system. However, for $T = 0.1$ s, 21 terms in the series for $C(z)$ are required to obtain the system response from $t = 0$ to $t = 2$ s. Hence the only practical technique of calculating the system response for this system is by simulation (which is usually the case).

6.3 SYSTEM CHARACTERISTIC EQUATION

Consider a single-loop sampled-data system of the type shown in Figure 6-5. For this system,

$$C(z) = \frac{G(z)R(z)}{1 + \overline{GH}(z)} = \frac{K \prod_{i=1}^m (z - z_i)}{\prod_{j=1}^n (z - p_j)} R(z)$$

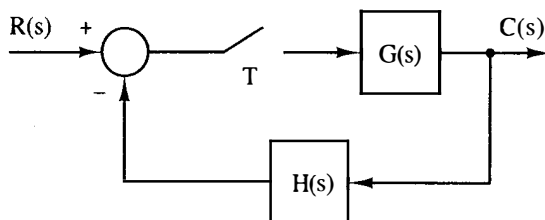


Figure 6-5 Single-loop sampled-data systems.

Using the partial-fraction expansion, we can express $C(z)$ as

$$C(z) = \frac{k_1 z}{z - p_1} + \cdots + \frac{k_n z}{z - p_n} + C_R(z) \quad (6-5)$$

where $C_R(z)$ contains the terms of $C(z)$ which originate in the poles of $R(z)$. The first n terms of (6-5) are the natural response terms of $C(z)$ [i.e., the terms that are always present in $C(z)$]. If the system is stable, these terms are also called the *transient response*. The inverse z -transform of the i th term yields

$$\mathcal{Z}^{-1} \left[\frac{k_i z}{z - p_i} \right] = k_i (p_i)^k$$

It is seen that these terms determine the nature, or character, of the system natural response. Since the p_i originate in the roots of the equation

$$1 + \overline{GH}(z) = 0$$

this equation is then the *system characteristic equation*. The roots of the characteristic equation are the poles of the closed-loop transfer function. If a transfer function cannot be written, the roots of the characteristic equation are the poles of $C(z)$ that are independent of the input function.

6.4 MAPPING THE s-PLANE INTO THE z-PLANE

In studying the characteristics of analog systems, we are able to assign time-response characteristics to closed-loop pole locations (characteristic-equation zero locations) [1]. It is desirable to be able to do the same for sampled-data systems, and this topic is discussed in this section.

To introduce the topic, consider a function $e(t)$, which is sampled with the resulting starred transform $E^*(s)$. At the sampling instants, the sampled signal is of the same nature (and has the same values) as the continuous signal. For example, if $e(t)$ is exponential, then the sampled signal is exponential at the sampling instants, with the same amplitude and time constant as the continuous function. If $e(t) = e^{-at}$,

$$E(s) = \frac{1}{s + a}, \quad E^*(s) = \frac{e^{Ts}}{e^{Ts} - e^{-aT}}, \quad E(z) = \frac{z}{z - e^{-aT}}$$

Hence an s -plane pole at $s = -a$ results in the z -plane pole at $z = e^{-aT}$. In general, from the z -transform tables of Appendix VIII, we see that a pole of $E(s)$ at $s = s_1$ results in a z -plane pole of $E(z)$ at $z_1 = e^{s_1 T}$. This characteristic is also evident from the second property of the starred transform given in Section 3.6 and from (4-4). We will use the inverse of this characteristic. A z -plane pole at $z = z_1$ results in the transient-response characteristics at the sampling instants of the equivalent s -plane pole s_1 , where s_1 and z_1 are related by $z_1 = e^{s_1 T}$.

Consider first the mapping of the left half-plane portion of the primary strip

into the z -plane as shown in Figure 6-6. Along the $j\omega$ axis,

$$z = e^{sT} = e^{\sigma T} e^{j\omega T} = e^{j\omega T} = \cos \omega T + j \sin \omega T = 1/\omega T$$

Hence poles located on the unit circle in the z -plane are equivalent to pole locations on the imaginary axis in the s -plane. Thus pole locations on the unit circle in the z -plane signify a system with a steady-state oscillation in its natural response. From the equation above, the frequency of oscillation is given by the angle of the pole (in radians) divided by T .

For $\omega = \omega_s/2$, ωT is equal to π , and hence the $j\omega$ axis between $-j\omega_s/2$ and $j\omega_s/2$ maps into the unit circle in the z -plane. In fact, any portion of the $j\omega$ axis of length ω_s maps into the unit circle in the z -plane. The right-half-plane portion of the primary strip (see Section 3.6) maps into the exterior of the unit circle, and the left-half-plane portion of the primary strip maps into the interior of the unit circle. Thus, since the stable region of the s -plane is the left half-plane, the stable region of the z -plane is the interior of the unit circle. Stability will be discussed in detail in Chapter 7.

Constant damping loci in the s -plane (i.e., straight lines with σ constant) map into circles in the z -plane as shown in Figure 6-7. This can be seen using the relationship

$$z = e^{\sigma_1 T} e^{j\omega T} = e^{\sigma_1 T} / \omega T$$

Constant frequency loci in the s -plane map into rays as shown in Figure 6-8.

For constant damping ratio loci, σ and ω are related by

$$\frac{\omega}{\sigma} = \tan \beta$$

where β is constant. Then

$$z = e^{sT} = e^{\sigma T} / \sigma T \tan \beta \quad (6-6)$$

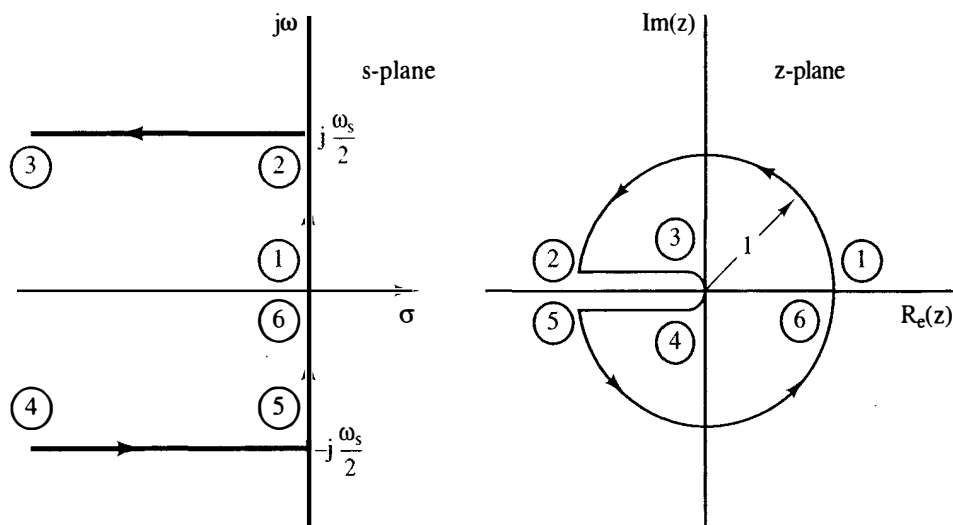


Figure 6-6 Mapping the primary strip into the z -plane.

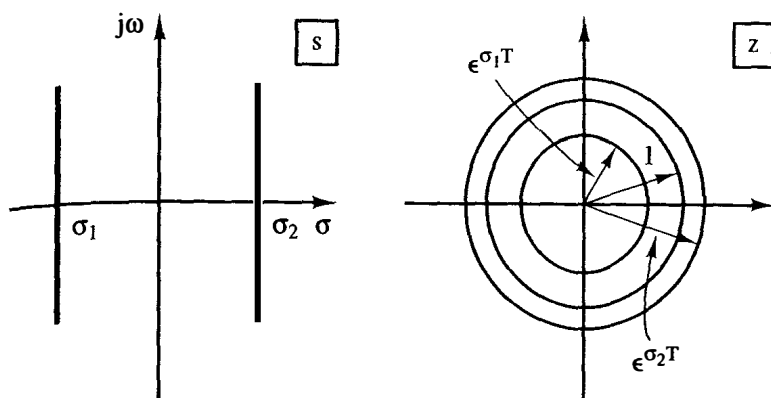


Figure 6-7 Mapping constant damping loci into the z-plane.

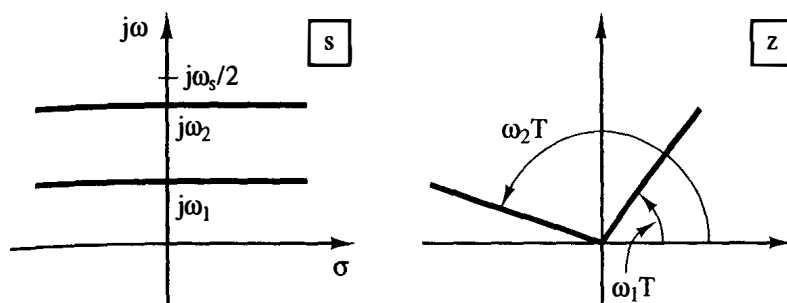


Figure 6-8 Mapping constant frequency loci into the z-plane.

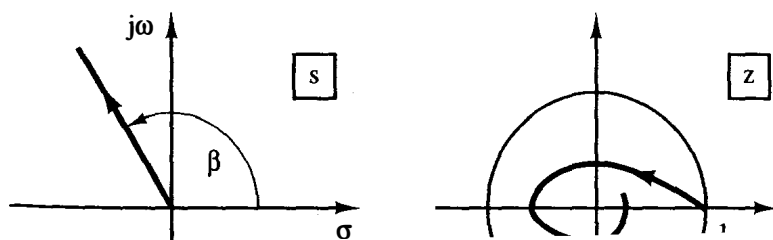
Since σ is negative in the second and third quadrants of the s -plane, (6-6) describes a logarithmic spiral whose amplitude decreases with σ increasing in magnitude. This is illustrated in Figure 6-9.

As described above, the characteristics of a sampled time function at the sampling instants are the same as those of the time function before sampling. Thus, using the mappings illustrated in Figures 6-6 through 6-9, we may assign time-response characteristics to characteristic-equation zero locations in the z -plane. The correspondence of several s -plane and z -plane pole locations is illustrated by several examples in Figure 6-10. The time-response characteristics of the z -plane pole locations are illustrated in Figure 6-11. Since

$$z = \epsilon^{sT} = \epsilon^{\sigma T} \epsilon^{j\omega T}$$

the time-response characteristics are a function not only of s , but also of T .

Consider the case in Figure 6-10 that s -plane poles occur at $s = \sigma \pm j\omega$. These poles result in a system transient-response term of the form $A \epsilon^{\sigma t} \cos(\omega t + \phi)$. When



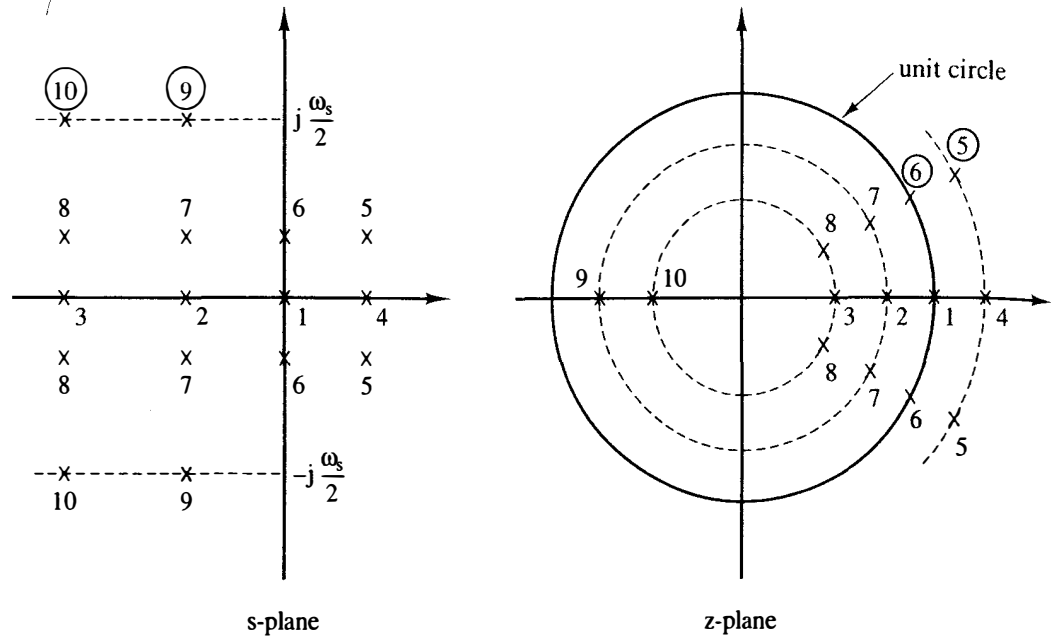


Figure 6-10 Corresponding pole locations between the s -plane and the z -plane.

sampling occurs, these s -plane poles result in z -plane poles at

$$z = e^{sT} \Big|_{s = \sigma \pm j\omega} = e^{\sigma T} e^{\pm j\omega T} = e^{\sigma T} \angle \pm \omega T = r \angle \pm \theta \quad (6-7)$$

Thus roots of the characteristic equation that appear at $z = r \angle \pm \theta$ result in a transient-response term of the form

$$A e^{\sigma k T} \cos(\omega k T + \phi) = A(r)^k \cos(\theta k + \phi)$$

Example 6.5

As an example, the time constant that appears in the sampled transient response of the first-order system of Example 6.1 will be calculated. The closed-loop transfer function for this system was found to be

$$\frac{G(z)}{1 + G(z)} = \frac{0.3625}{z - 0.4562}$$

Hence the closed-loop characteristic equation is

$$z - 0.4562 = 0$$

From Figure 6-10, we see that a pole at $z = 0.4562$ corresponds to an s -plane pole s_1 on the negative real axis that satisfies

$$z_1 = 0.4562 = e^{s_1 T} = e^{0.1 s_1}$$

Hence $s_1 = \ln(0.4562)/0.1$, or $s_1 = -7.848$. Since the time constant is the reciprocal of the magnitude of a real pole in the s -plane, the closed-loop system has the time constant $\tau = 0.127$ s. (From Example 6.3 the time constant of the system with the sampling removed is 0.167 s.) Also, since the transient response settles out in approximately four time constants, the transient response of this system will settle in approximately 0.5 s. This characteristic is seen in the step response of Figure 6-1b.

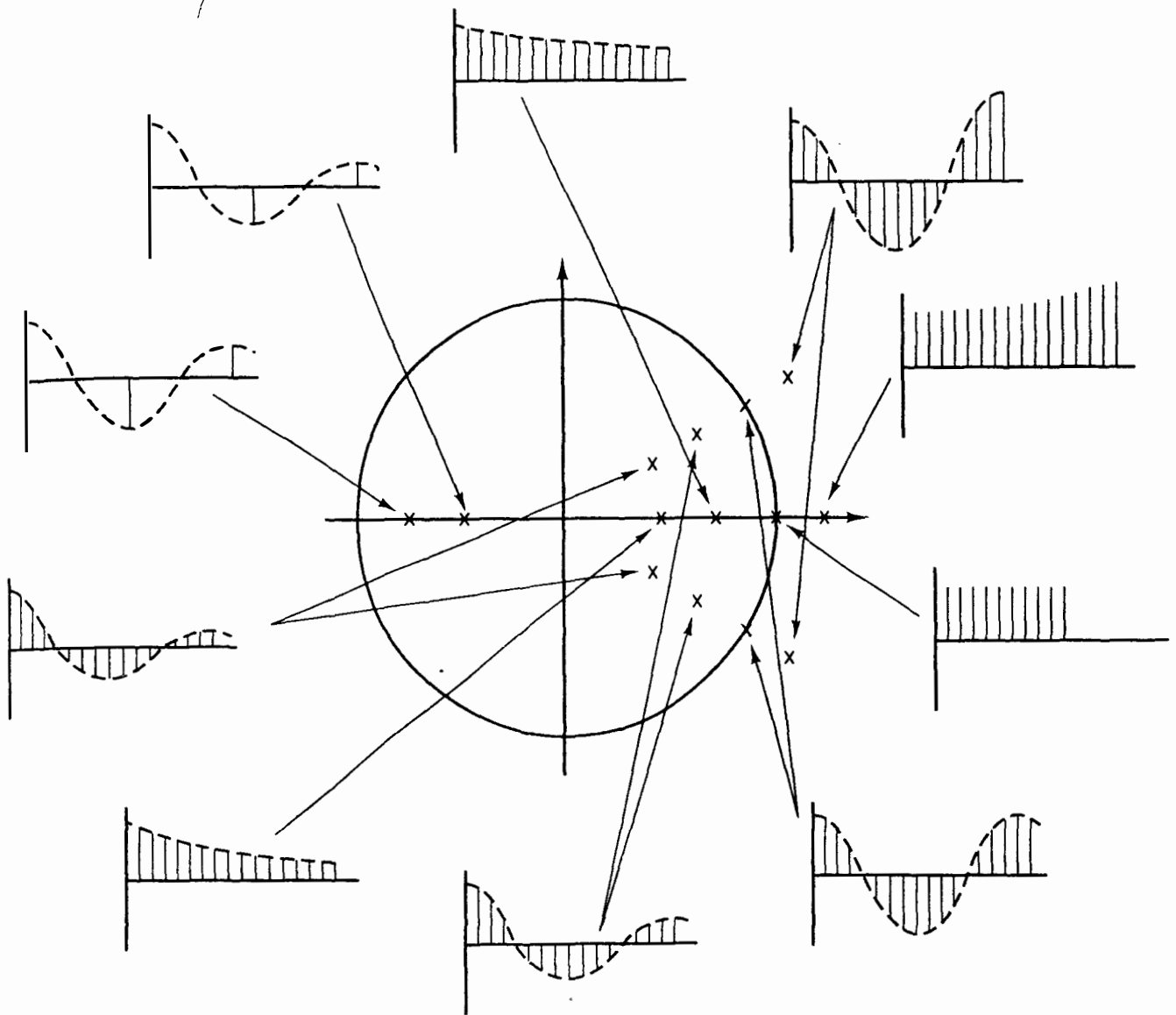


Figure 6-11 Transient response characteristics of the z-plane pole locations.

In the discussion above, we considered the relationship between s-plane poles and z-plane poles in a general way. We will now mathematically relate the s-plane pole locations and the z-plane pole locations. We express in standard form the s-plane second-order transfer function

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

which has the poles

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n \sqrt{1 - \zeta^2}$$

where ζ is the damping ratio and ω_n is the natural frequency. The equivalent z-plane poles occur at

$$z = e^{sT}|_{s_{1,2}} = e^{-\zeta\omega_n T} / \pm \omega_n T \sqrt{1 - \zeta^2} = r / \pm \theta$$

Hence

$$\epsilon^{-\zeta\omega_n T} = r$$

or

$$\zeta\omega_n T = -\ln r$$

Also,

$$\omega_n T \sqrt{1 - \zeta^2} = \theta$$

Taking the ratio of the last two equations, we obtain

$$\frac{\zeta}{\sqrt{1 - \zeta^2}} = \frac{-\ln r}{\theta}$$

Solving this equation for ζ yields

$$\zeta = -1 \quad \frac{\ln r}{\ln r} \quad \theta = \theta \quad \zeta = \frac{-\ln r}{\sqrt{\ln^2 r + \theta^2}} \quad (6-8)$$

We then find ω_n to be

$$\frac{\ln r}{F} \quad \omega_n = \frac{1}{T} \sqrt{\ln^2 r + \theta^2} \quad (6-9)$$

The time constant, τ , of the poles is then given by

$$\tau = \frac{1}{\zeta\omega_n} = \frac{-T}{\ln r} \quad (6-10)$$

This equation can also be expressed as

$$r = \epsilon^{-T/\tau}$$

Thus, given the complex pole location in the z -plane, we find the damping ratio, the natural frequency, and the time constant of the pole from (6-8), (6-9), and (6-10), respectively.

Example 6.6

For this example we will consider the system of Example 6.4. For this system the closed-loop transfer function was calculated to be

$$\frac{G(z)}{1 + G(z)} = \frac{0.368z + 0.264}{z^2 - z + 0.632}, \quad T = 1 \text{ s}$$

Thus the system characteristic equation is

$$z^2 - z + 0.632 = (z - 0.5 - j0.618)(z - 0.5 + j0.618) = 0$$

The poles are then complex and occur at

$$z = 0.5 \pm j0.618 = 0.795/\pm 51.0^\circ = 0.795/\pm 0.890 \text{ rad}$$

Since, in (6-8), (6-9), and (6-10),

$$z = e^{\sigma T} / \pm \omega T = r / \pm \omega T = 0.795 / \pm 0.890$$

then

$$\zeta = \frac{-\ln(0.795)}{[\ln^2(0.795) + (0.890)^2]^{1/2}} = 0.250$$

$$\omega_n = \frac{1}{T} [\ln^2(0.795) + (0.890)^2]^{1/2} = 0.9191$$

$$\tau = \frac{-1}{\ln(0.795)} = 4.36 \text{ s}$$

The equivalent values for the closed-loop analog system (with sampler and data hold removed) were found in Example 6.4 to be $\zeta = 0.50$ and $\omega_n = 1$ rad/s. Also $\tau = 1/\zeta\omega_n = 2$ s. Hence the effects of the sampling are seen to be destabilizing. However, if $T = 0.1$ s, there is little effect from the sampling. From Problem 6.10, the parameters for this case are $\zeta = 0.475$, $\omega_n = 0.998$ rad/s, and $\tau = 2.11$ s.

A word is in order concerning the probable location of characteristic-equation zeros. Transfer-function pole locations in the s-plane transform change into z-plane pole locations as

$$s + 1/\tau \rightarrow z - e^{-T/\tau}$$

$$(s + 1/\tau)^2 + \omega^2 \rightarrow z^2 - 2ze^{-T/\tau} \cos \omega T + e^{-2T/\tau} = (z - z_1)(z - \bar{z}_1)$$

where

$$z_1 = e^{-T/\tau} e^{j\omega T} = e^{-T/\tau} / \omega T = r / \theta$$

The s-plane time constant for the real pole is τ . In order for the sampling to have negligible effect, T must be much less than τ . Thus for the real pole, the z-plane pole location will be in the vicinity of $z = 1$, since T/τ is much less than 1. For the complex pole, an additional requirement is that several samples be taken per cycle of the sinusoid, or that $\omega T < 1$. Thus, once again, $T \ll \tau$, and since $z_1 \bar{z}_1 = e^{-2T/\tau}$, then $|z_1| = e^{-T/\tau}$ and the z-plane pole locations again will be in the vicinity of $z = 1$. In general, for a discrete-time control system, the transfer-function pole locations (characteristic-equation zero locations) are placed in the vicinity of $z = 1$, if system constraints allow a sufficiently high sample rate to be chosen.

The ideas in the foregoing paragraph can be expressed mathematically. From (6-10),

$$\frac{\tau}{T} = -\frac{1}{\ln r} \quad (6-11)$$

We see that the ratio τ/T is simply the number of samples per time constant. For example, if $\tau = 1$ s and $T = 0.25$ s, then we have four samples per time constant. Equation (6-11) is tabulated for certain values of r in Table 6-2.

TABLE 6-2

Samples per time constant		Samples per period	
r	τ/T	θ	T_d/T
0.999	999.5	10°	36
0.99	99.5	20°	18
0.95	19.5	30°	12
0.9	9.5	45°	8
0.8	4.48	60°	6
0.7	2.80	90°	4
0.6	1.96	120°	3
0.4	1.09	180°	2
0.2	0.62		

In a like manner, we can manipulate the equation $\omega T = \theta$ to obtain the number of samples per cycle of the sinusoid. Thus

$$\omega T = \frac{2\pi}{T_d} T = \theta$$

where T_d is the period of the sinusoid and θ is in radians. Hence

$$\frac{T_d}{T} = \frac{2\pi}{\theta} = \frac{360^\circ}{\theta^\circ} \quad (6-12)$$

where θ° denotes θ in degrees. Equation (6-12) is also tabulated for certain values of θ in Table 6-2.

6.5 STEADY-STATE ACCURACY

An important characteristic of a control system is its ability to follow, or track, certain inputs with a minimum of error. The control system designer attempts to minimize the system error to certain anticipated inputs. In this section the effects of the system transfer characteristics on the steady-state system errors are considered.

Consider the system of Figure 6-12. For this system,

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1 + G(z)} \quad (6-13)$$

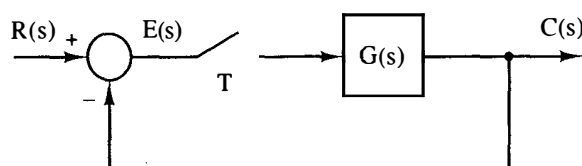


Figure 6-12 Discrete-time system.

where $G(z) = \mathcal{Z}[G(s)]$. The plant transfer function can always be expressed as

$$G(z) = \frac{K \prod_{i=1}^m (z - z_i)}{(z - 1)^N \prod_{j=1}^p (z - z_j)}, \quad z_i \neq 1, \quad z_j \neq 1 \quad (6-14)$$

As we shall see, the value of N has special significance and is called the *system type*. For convenience in the following development, we define

$$K_{dc} = \left. \frac{K \prod_{i=1}^m (z - z_i)}{\prod_{j=1}^p (z - z_j)} \right|_{z=1} \quad (6-15)$$

Note that K_{dc} is the open-loop plant dc gain with all poles at $z = 1$ removed.

For the system of Figure 6-12, the system error, $e(t)$, is defined as the difference between the system input and the system output. Or

$$E(z) = \mathcal{Z}[e(t)] = R(z) - C(z) \quad (6-16)$$

Then, from (6-13) and (6-16),

$$E(z) = R(z) - \frac{G(z)}{1 + G(z)} R(z) = \frac{R(z)}{1 + G(z)}$$

The steady-state errors will now be derived for two common inputs—a position (step) input and a velocity (ramp) input. First, for the unit-step input,

$$R(z) = \frac{z}{z - 1}$$

Then, from the final-value theorem, the steady-state error is seen to be

$$e_{ss}(kT) = \lim_{z \rightarrow 1} (z - 1)E(z) = \lim_{z \rightarrow 1} \frac{(z - 1)R(z)}{1 + G(z)} \quad (6-17)$$

provided that $e_{ss}(kT)$ has a final value. The steady-state error is then

$$e_{ss}(kT) = \lim_{z \rightarrow 1} \frac{z}{1 + G(z)} = \frac{1}{1 + \lim_{z \rightarrow 1} G(z)}$$

We now define the position error constant as

$$K_p = \lim_{z \rightarrow 1} G(z) \quad (6-18)$$

Then in (6-14), if $N = 0$ [i.e., no poles in $G(z)$ at $z = 1$], $K_p = K_{dc}$ and

$$e_{ss}(kT) = \frac{1}{1 + K_p} = \frac{1}{1 + K_{dc}} \quad (6-19)$$

For $N \geq 1$ (system type greater than or equal to one), $K_p = \infty$ and the steady-state error is zero.

Consider next the unit-ramp input. In this case $r(t) = t$, and from Appendix VIII,

$$R(z) = \frac{Tz}{(z-1)^2}$$

Then, from (6-17),

$$e_{ss}(kT) = \lim_{z \rightarrow 1} \frac{Tz}{(z-1) + (z-1)G(z)} = \lim_{z \rightarrow 1} \frac{T}{(z-1)G(z)}$$

We now define the velocity error constant as

$$K_v = \lim_{z \rightarrow 1} \frac{1}{T} (z-1)G(z) \quad (6-20)$$

Then if $N = 0$, $K_v = 0$ and $e_{ss}(kT) = \infty$. For $N = 1$, $K_v = K_{dc}/T$ and

$$e_{ss}(kT) = \frac{1}{K_v} = \frac{T}{K_{dc}} \quad (6-21)$$

For $N \geq 2$ (system type greater than or equal to 2), $K_v = \infty$ and $e_{ss}(kT)$ is zero.

The development above illustrates that, in general, increased system gain and/or the addition of poles at $z = 1$ to the open-loop forward-path transfer function tend to decrease steady-state errors. However, as will be demonstrated in Chapter 7, both large gains and poles of $G(z)$ at $z = 1$ have destabilizing effects on the system. Generally, trade-offs exist between small steady-state errors and adequate system stability (or acceptable system transient response).

Example 6.7

The steady-state errors will be calculated for the system of Figure 6-12, in which the open-loop function is given as

$$G(s) = \frac{1 - \epsilon^{-Ts}}{s} \left[\frac{K}{s(s+1)} \right]$$

Thus

$$\begin{aligned} G(z) &= K \mathcal{Z} \left[\frac{1 - \epsilon^{-Ts}}{s^2(s+1)} \right] = \frac{K(z-1)}{z} \mathcal{Z} \left[\frac{1}{s^2(s+1)} \right] \\ &= \frac{K(z-1)}{z} \frac{z[(\epsilon^{-T} + T - 1)z + (1 - \epsilon^{-T} - T\epsilon^{-T})]}{(z-1)^2(z - \epsilon^{-T})} \\ &= \frac{K[(\epsilon^{-T} + T - 1)z + (1 - \epsilon^{-T} - T\epsilon^{-T})]}{(z-1)(z - \epsilon^{-T})} \end{aligned}$$

Then, from (6-14) and (6-20), the system is type 1 and

$$K_v = \frac{K_{dc}}{T} = \frac{K[(\epsilon^{-T} + T - 1) + (1 - \epsilon^{-T} - T\epsilon^{-T})]}{T(1 - \epsilon^{-T})} = K$$

Since $G(z)$ has one pole at $z = 1$, the steady-state error to a step input is zero, and to

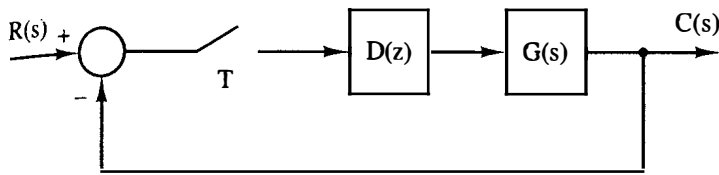


Figure 6-13 System for Example 6.8.

a ramp input is, from (6-21),

$$e_{ss}(kT) = \frac{1}{K_v} = \frac{1}{K}$$

provided that the system is stable. The question of stability is considered in Chapter 7.

Example 6.8

As a second example, consider again the system of Figure 6-12, where, for this example,

$$G(z) = \mathcal{Z} \left[\frac{1 - e^{-Ts}}{s(s+1)} \right] = \frac{1 - e^{-T}}{z - e^{-T}}$$

Suppose that the design specification for this system requires that the steady-state error to a unit ramp input be less than 0.01. Thus, from (6-20), it is necessary that the open-loop system be type 1 or greater and thus the open-loop function must have at least one pole at $z = 1$. Since the plant does not contain a pole at $z = 1$, a digital compensator of the form

$$D(z) = \frac{K_I z}{z - 1} + K_P$$

will be added, to produce the resultant system shown in Figure 6-13. The compensator, called a PI or proportional-plus-integral compensator, is of a form commonly used to reduce steady-state errors. This compensator is discussed in Chapter 8. For this system (6-20) becomes

$$K_v = \lim_{z \rightarrow 1} \frac{1}{T} (z - 1) D(z) G(z)$$

Employing the expressions above for $D(z)$ and $G(z)$, we see that

$$K_v = \lim_{z \rightarrow 1} (z - 1) \frac{(K_I + K_P)z - K_P}{T(z - 1)} \left[\frac{1 - e^{-T}}{z - e^{-T}} \right] = \frac{K_I}{T}$$

Thus K_I equals $100T$ for the required steady-state error, provided that the system is stable. The latter point is indeed an important consideration since the error analysis is meaningless unless stability of the system is guaranteed. Chapter 7 illustrates a number of techniques for analyzing system stability. This system is considered further in Example 7.6.

6.6 SIMULATION

Thus far we have determined the time response of a system via a transform approach. In Example 6.1 we determined the system response at the sampling instants using

the z -transform. A Laplace transform technique for calculating the system response for all time was also illustrated, but was discarded for being too unwieldy.

A different approach for determining a system's response is through simulation. The simulation of a continuous-time (analog) system may be via the integration of the system's differential equations using electronic circuits. The analog computer is designed to perform this function. The interconnection of a digital computer with the analog computer is called a hybrid computer and is useful in simulating digital control systems. The analog parts of the control system are simulated on the analog computer with the digital controller simulated on the digital computer.

If a numerical algorithm is used to integrate an analog system's differential equations on a digital computer, we then have a digital simulation of the system. This is the type of simulation that we consider here.

The problem of numerical integration of a time function can be illustrated using Figure 6-14. We wish to integrate $y(t)$ numerically; that is, we wish to find $x(t)$, where

$$x(t) = \int_0^t y(\tau) d\tau + x(0) \quad (6-22)$$

Suppose that we know $x[(k-1)H]$ and we want to calculate $x(kH)$, where H is called the *numerical integration increment* and is the step size in the algorithm. Perhaps the simplest numerical integration algorithm is obtained by assuming $y(t)$ constant at the value $y[(k-1)H]$ for $(k-1)H \leq t < kH$. Then

$$x(kH) = x[(k-1)H] + Hy[(k-1)H] \quad (6-23)$$

Of course, $x(kH)$ in (6-23) is only an approximation to $x(t)$ in (6-22) evaluated at $t = kH$. Note in Figure 6-14 that we are approximating the area under the $y(t)$ curve for $(k-1)H \leq t < kH$ with the area of the rectangle shown. For this reason, this numerical integration algorithm is called the *rectangular rule*, and is also known as the *Euler method*.

We are interested in the integration of differential equations [i.e., in (6-22), $y(t) = \dot{x}(t)$]. We will illustrate this case with a simple example. Suppose that we wish to solve the differential equation

$$\dot{x}(t) + x(t) = 0, \quad x(0) = 1 \quad (6-24)$$

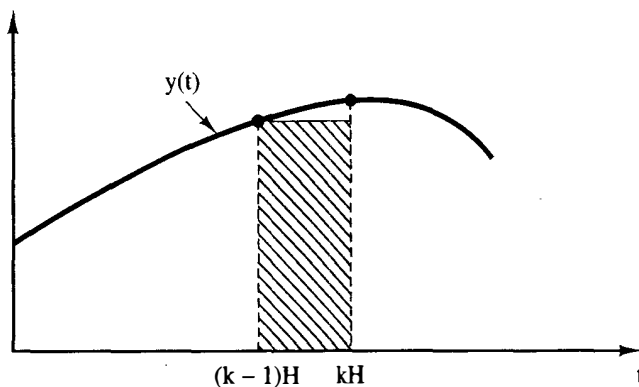


Figure 6-14 Rectangular rule for numerical integration.

The solution is obviously

$$x(t) = e^{-t}$$

However, by numerical integration in (6-23),

$$x(kH) = x[(k-1)H] + H[-x[(k-1)H]] \quad (6-25)$$

since, from (6-24), for this differential equation

$$y(t) = \dot{x}(t) = -x(t)$$

Suppose that we chose $H = 0.1$ s. Then, solving (6-25) iteratively starting with $k = 1$ [we know $x(0)$],

$$x(0.1) = x(0) - Hx(0) = 1.0 - (0.1)(1.0) = 0.9$$

$$x(0.2) = x(0.1) - Hx(0.1) = 0.9 - 0.09 = 0.81$$

$$\vdots$$

$$x(1.0) = x(0.9) - Hx(0.9) = 0.3487$$

Since, for this example, we know the solution, we calculate $x(t)$ at $t = 1.0$ s as

$$x(1.0) = e^{-1.0} = 0.3678$$

and we can see the error due to the numerical integration.

In the example above, if we choose H larger, the error is larger. If we decrease H , the error decreases to a minimum value. Then a further decrease in H will result in an increase in the error, due to round-off in the computer. With H equal to 0.1 s, 10 iterations are required to calculate $x(1)$. With $H = 0.001$ s, 1000 iterations are required to calculate $x(1)$. The round-off errors in the computer are larger for the latter case, since more calculations are made. If H is made sufficiently smaller, this round-off error becomes appreciable.

Next we will consider the simulation of an analog system. Assume that the system's state equations are given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c \mathbf{x}(t) + \mathbf{B}_c \mathbf{u}(t) \quad (6-26)$$

Then (6-22) becomes

$$\mathbf{x}(t) = \int_0^t \dot{\mathbf{x}}(t) dt + \mathbf{x}(0) \quad (6-27)$$

and the rectangular rule for $\mathbf{x}(t)$ a vector becomes

$$\mathbf{x}(kH) = \mathbf{x}[(k-1)H] + H\dot{\mathbf{x}}[(k-1)H] \quad (6-28)$$

where, from (6-26),

$$\dot{\mathbf{x}}[(k-1)H] = \mathbf{A}_c \mathbf{x}[(k-1)H] + \mathbf{B}_c \mathbf{u}[(k-1)H] \quad (6-29)$$

The numerical integration algorithm becomes:

1. Let $k = 1$.
2. Evaluate $\dot{x}[(k - 1)H]$ in (6-29).
3. Evaluate $x(kH)$ in (6-28).
4. Let $k = k + 1$.
5. Go to step 2.

Refer again to Figure 6-14. This figure illustrates the rectangular rule for numerical integration. Other numerical integration algorithms differ from this rule in the manner that $x[kH]$ is calculated, based on the knowledge of $\dot{x}[(k - 1)H]$. As an example of a different algorithm, consider the trapezoidal rule illustrated in Figure 6-15. The integral for $[(k - 1)H] \leq t < kH$ is approximated by the area of the trapezoid. For this rule, with $x(t)$ equal to the integral of $y(t)$,

$$x(kH) = x[(k - 1)H] + H \left[\frac{y[(k - 1)H] + y(kH)}{2} \right] \quad (6-30)$$

For $y(t)$ equal to $\dot{x}(t)$, then

$$x(kH) = x[(k - 1)H] + \frac{H}{2} [\dot{x}[(k - 1)H] + \dot{x}(kH)] \quad (6-31)$$

Since $x[(k - 1)H]$ is known, $\dot{x}[(k - 1)H]$ can be calculated from the differential equation that is to be solved. However, we cannot calculate $\dot{x}(kH)$ without a knowledge of $x(kH)$. A method for solving this problem will now be presented.

We use a different rule to "predict" the value of $x(kH)$. Using this predicted value, we calculate the predicted value of $\dot{x}(kH)$, and substitute this value in the trapezoidal rule to "correct" the value of $x(kH)$. A method of this type is called a predictor-corrector algorithm. A commonly used rule for prediction is the rectangular rule.

The state model of an analog system given in (6-26) will be used to illustrate the trapezoidal rule. It is assumed that $x[kH]$ is known for $k = 0$, and thus $x[kH]$ is to be calculated, for $k = 1, 2, 3, \dots$. The predictor algorithm, which is the rectangular rule, is given by

$$\dot{x}[(k - 1)H] = A_c x[(k - 1)H] + B_c u[(k - 1)H] \quad (6-32)$$

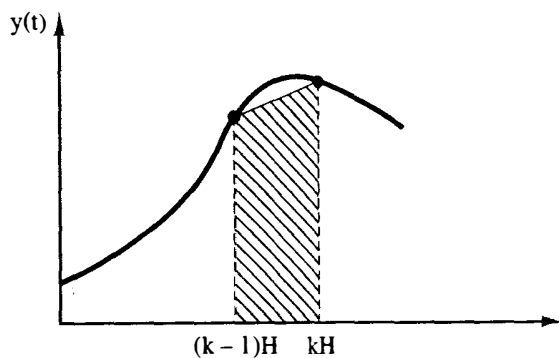


Figure 6-15 Trapezoidal rule for numerical integration.

$$\mathbf{x}(kH) = \mathbf{x}[(k-1)H] + H\dot{\mathbf{x}}[(k-1)H] \quad (6-33)$$

The corrector algorithm is given by

$$\dot{\mathbf{x}}(kH) = \mathbf{A}_c \mathbf{x}(kH) + \mathbf{B}_c \mathbf{u}(kH) \quad (6-34)$$

$$\mathbf{x}(kH) = \mathbf{x}[(k-1)H] + \frac{H}{2} [\dot{\mathbf{x}}[(k-1)H] + \dot{\mathbf{x}}(kH)] \quad (6-35)$$

The predicted value of $\mathbf{x}(kH)$ is given by (6-33), and the corrected, and final, value of $\mathbf{x}(kH)$ is given by (6-35).

Thus the numerical integration algorithm is given by:

1. Let $k = 1$.
2. Evaluate $\dot{\mathbf{x}}[(k-1)H]$ in (6-32).
3. Evaluate $\mathbf{x}(kH)$ in (6-33).
4. Evaluate $\dot{\mathbf{x}}(kH)$ in (6-34), using the result of step 3.
5. Evaluate the final value of $\mathbf{x}(kH)$ in (6-35).
6. Let $k = k + 1$.
7. Go to step 2.

This algorithm obviously requires more calculations per iteration than does the rectangular rule. However, for a specified accuracy of the solution, a much larger increment H may be used such that the total computer execution time for the trapezoidal rule is less than that for the rectangular rule. The trapezoidal rule will now be illustrated by an example.

Example 6.9

Consider once again the differential equation

$$\dot{x}(t) + x(t) = 0, \quad x(0) = 1$$

which has the solution

$$x(t) = e^{-t}$$

Since

$$\dot{x}(t) = -x(t)$$

implementing the trapezoidal rule for $H = 0.1$ s yields

$$K = 1:$$

$$\dot{x}(0) = -x(0) = -1$$

$$x(0.1) = x(0) + H\dot{x}(0) = 1 + 0.1(-1) = 0.9$$

$$\dot{x}(0.1) = -x(0.1) = -0.9$$

$$\begin{aligned} x(0.1) &= x(0) + H/2[\dot{x}(0) + \dot{x}(0.1)] \\ &= 1 + 0.05(-1 - 0.9) = 0.905 \end{aligned}$$

$K = 2$:

$$\dot{x}(0.1) = -x(0.1) = -0.905$$

$$x(0.2) = x(0.1) + H\dot{x}(0.1) = 0.905 - 0.0905 = 0.8145$$

$$\dot{x}(0.2) = -x(0.2) = -0.8145$$

$$\begin{aligned} x(0.2) &= x(0.1) + H/2[\dot{x}(0.1) + \dot{x}(0.2)] \\ &= 0.905 + 0.05(-0.905 - 0.8145) = 0.8190 \end{aligned}$$

for the first two iterations. If these calculations are continued, we find $x(1.0)$ to be 0.3685. Recall that the exact value of $x(1)$ is 0.3678, and the value of $x(1)$ calculated above using the rectangular rule is 0.3487. Hence much greater accuracy results from use of the trapezoidal rule as compared to the rectangular rule, but at the expense of more calculations. If H is increased to 0.333 s, $x(1)$ is calculated to be 0.3767 with the trapezoidal rule. Here the accuracy is still significantly greater than that of the rectangular rule with $H = 0.1$ s, although the amount of calculations required is approximately the same.

In the discussion above, only two of many numerical integration rules were discussed. One of the more commonly used rules for digital simulation is the fourth-order Runge-Kutta rule [2-4]. The interested reader is referred to the many texts available in this area.

Two additional points concerning simulation should be made. First, in digital simulations we approximate differential equations by difference equations and solve the difference equations. Thus we are replacing a continuous-time system with a discrete-time system that has approximately the same response.

The second point is that, in the simulation of nonlinear systems, the nonlinearities appear only in the calculation of $\dot{x}[(k-1)H]$, given $x[(k-1)H]$. If the nonlinearities are of a form that can be easily expressed in a mathematical form, the simulation is not appreciably more difficult to write.

In the discussion above, we considered the simulation of analog systems only. The addition of a sampler and zero-order hold to an analog system requires that logic be added to the system simulation. The logic will hold the zero-order-hold output constant over any sample period, and equal to the value of sampler input at the beginning of that sample period. The addition of a digital controller requires that the controller difference equation be solved only at the beginning of each sample period, and that the controller output then remain constant over the sample period.

6.7 CONTROL SOFTWARE

Many commercial analysis and design control software packages are available; these packages have simulation capabilities of varying degrees. The software packages CTRL and CSP are discussed in Appendix VI. CSP has limited capabilities, while CTRL, which is based on MATLAB, is of a more general nature. These packages are typical of those available.

In general, the analog components in a digital control system are entered into the software as transfer functions or state equations. Most packages will accept either model and then calculate the other model from the one given. If the digital controller is known, it is entered as a transfer function or, in some cases, as a state model. The packages can then simulate the system. In general, the packages have a number of numerical integration algorithms available; the user must choose one of these algorithms. For example, both CTRL and CSP contain the Euler rule and the fourth-order Runge-Kutta algorithm (a commonly-used algorithm in control).

6.8 SUMMARY

The time response of discrete-time closed-loop systems has been discussed. Both steady-state and transient responses have been considered. The correlation between the s -plane and the z -plane which has been presented provides a mechanism for the transfer of many of the continuous-system tools needed for both the analysis and design of closed-loop discrete-time systems. Finally, a brief introduction into the digital simulation of systems is presented.

REFERENCES AND FURTHER READING

1. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2d ed. Englewood Cliffs, NJ: Prentice Hall, 1991.
2. M. L. Dertouzos et al., *Systems, Networks, and Computation: Basic Concepts*. New York: McGraw-Hill Book Company, 1972.
3. J. L. Melsa, *Computer Programs for Computational Assistance*. New York: McGraw-Hill Book Company, 1972.
4. C. F. Gerald, *Applied Numerical Analysis*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1984.
5. J. A. Cadzow and H. R. Martens, *Discrete-Time and Computer Control Systems*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1970.
6. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1988.
7. E. I. Jury, *Theory and Application of the z -Transform Method*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.
8. B. C. Kuo, *Digital Control Systems*, 2d ed. New York: Saunders College Publishing, 1992.

PROBLEMS

6-1. Consider the closed-loop system of Figure P6-1.

- (a) Calculate and plot the unit-step response at the sampling instants, for the case that $D(z) = 1$.
- (b) Calculate the system unit-step response of the analog system, that is, with the

sampler, digital controller, and data hold removed. Plot the response on the same graph with the results of part (a).

- (c) For the system of Figure P6-1, let $D(z) = 1$ and $T = 0.4$ s. Calculate the unit-step response and plot these results on the same graph used for parts (a) and (b).
- (d) Use the system dc gains to calculate the steady-state responses for each of the systems of parts (a), (b), and (c). Why are these gains equal?
- (e) If simulation facilities are available, run the unit-step response for each system described in this problem and compare these responses to the calculated responses.

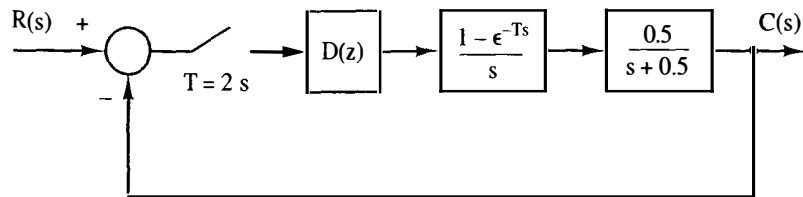


Figure P6-1 System for Problem 6-1.

- 6-2. Consider the system of Figure P6-1, with $D(z) = 1$. Use the results of Problem 6-1 if available.
 - (a) Find the system time constant τ for $T = 2$ s.
 - (b) With the input a step function, find the time required for the system output $c(kT)$ to reach 98 percent of its final value, for $T = 2$ s. Recall that four time constants (4τ) are required.
 - (c) Repeat parts (a) and (b) for $T = 0.4$ s.
 - (d) Repeat parts (a) and (b) for the analog system, that is, for the system with the sampler-data-hold removed.
- 6-3. In Example 6.1 the response of a sampled-data system between sample instants was expressed as a sum of delayed step responses.
 - (a) Use this procedure to find the system output $y(t)$ of Figure P6-1 at $t = 1$ s.
 - (b) Repeat part (a) at $t = 3$ s.
 - (c) The equation for $c(t)$ in part (a) and that for $c(t)$ in part (b) should give the same value at $t = 2$ s. Why? *Hint:* Consider the step response of the analog plant.
 - (d) Show that the statement in part (c) is true.
- 6-4. Shown in Figure P6-4 is the block diagram of a temperature control system for a large test chamber. This system is described in Problem 1-10. Ignore the disturbance input for this problem.
 - (a) With $D(z) = 1$ and $T = 0.6$ s, evaluate and plot the system response if the input is to command a 10°C step in the output. Note that the system input must be a step function with an amplitude of 0.4 V. Why?
 - (b) Use the results of part (a) to plot the output of the zero-order hold.
 - (c) Solve for the steady-state output for part (a).
 - (d) Suppose that the gain of 2 in the plant is replaced with a variable gain K . What value does the output approach in the steady state as K becomes very large? Assume that the system remains stable as K is increased (an unrealistic assumption).

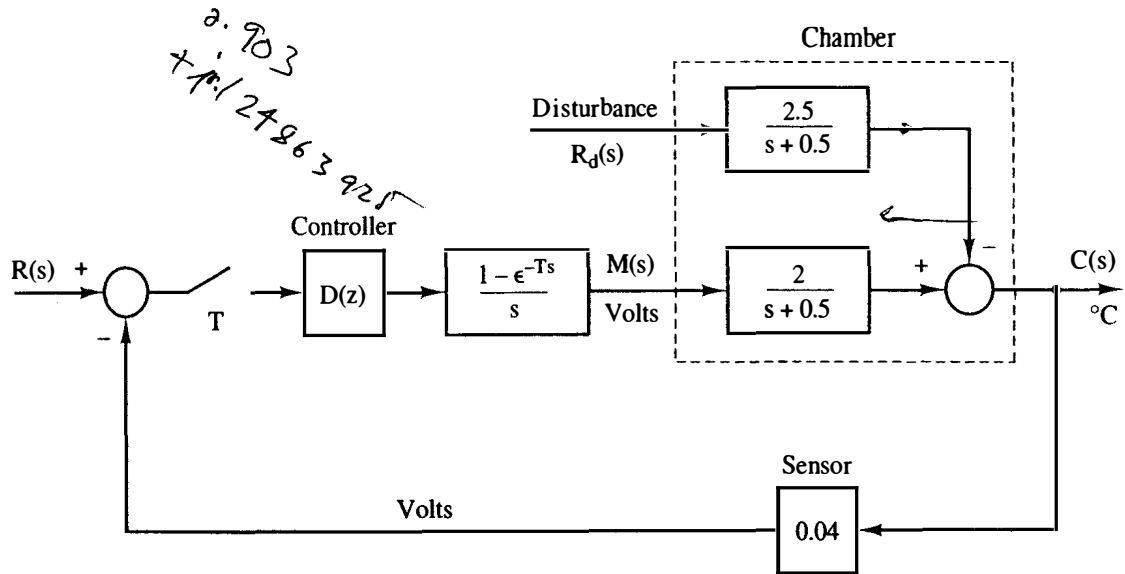


Figure P6-4 Chamber temperature control system.

6-5. Consider the temperature control system of Problem 6-4 and Figure P6-4.

- (a) Let $T = 6$ s, and solve for the response to the input $R(s) = 0.4$ s $^{-1}$. Plot this response on the same graph with the response found in Problem 6-4. Note the effects of increasing T from 0.6 s to 6 s, when the plant has a time constant of 2 s.
- (b) Find $c(t)$ for $0 \leq t \leq 6$ s. This response can be calculated without the use of the z -transform.

6-6. Consider the system of Figure P6-4, with $D(z) = 1$. Use the results of Problems 6-4 and 6-5 if available.

- (a) Find the system time constant τ for $T = 0.6$ s.
- (b) With the input a step function, find the time required for the system output $c(kT)$ to reach 98 percent of its final value. Note that this time is approximately four time constants (4τ).
- (c) Repeat parts (a) and (b) for $T = 6$ s.
- (d) Repeat parts (a) and (b) for the analog system, that is, for the system with the sampler/data hold removed.

6-7. The block diagram of a control system of a joint in a robot arm is shown in Figure P6-7. This system is discussed in Section 1.6. Let $T = 0.1$ s and $D(z) = 1$.

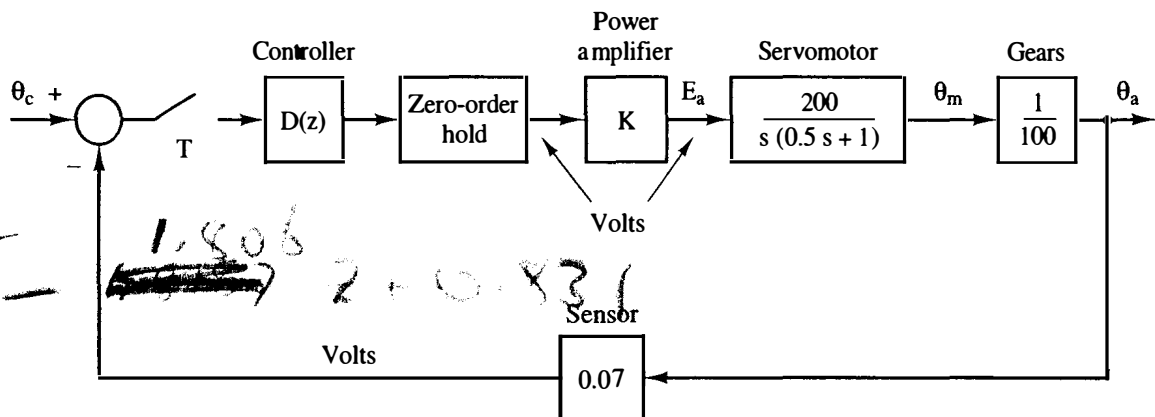


Figure P6-7 Robot arm joint control system.

- (a) Evaluate $C(z)$ if the input is to command a 20% step in the output and $K = 10$. Note that the system input must be a step function with an amplitude 1.4 V. Why?
- (b) Assuming the system to be stable, find the steady-state system output.
- (c) Find the (approximate) time required for the system response to reach steady state.
- (d) Simulate the system to verify the results in parts (b) and (c).
- 6-8.** The block diagram of a control system of a joint in a robot arm is shown in Figure P6-7. Let $T = 0.1$ s, $K = 10$, and $D(z) = 1$. The results of Problem 6-7 are useful in this problem if these results are available.
- (a) Find the damping ratio ζ , the natural frequency ω_n , and the time constant τ of the open-loop system. If the system characteristic equation has two real zeros, find the two time constants. These values can be solved by inspection. Why?
- (b) Repeat part (a) for the closed-loop system.
- (c) Repeat parts (a) and (b) for the system with the sampler, digital controller, and data hold removed, that is, for the analog system.
- (d) Use the results in parts (b) and (c) to find the percent overshoot in the step responses for the sampled-data closed-loop system and for the analog closed-loop system.
- 6-9.** The block diagram of an attitude control system of a satellite is shown in Figure P6-9. Let $T = 1$ s, $K = 100$, $J = 0.1$, $H_k = 0.02$, and $D(z) = 1$.
- (a) Find the damping ratio ζ , the natural frequency ω_n , and the time constant τ of the open-loop system. If the system characteristic equation has two real zeros, find the two time constants. These values can be solved by inspection. Why?
- (b) Repeat part (a) for the closed-loop system.
- (c) Repeat parts (a) and (b) for the system with the sampler, digital controller, and data hold removed, that is, for the analog system.
- (d) The closed-loop sampled-data system is seen to be unstable and the closed-loop analog system is seen to be marginally stable. If the satellite is operated with each of these control systems, describe the resulting movement for both the sampled-data system and the analog system.

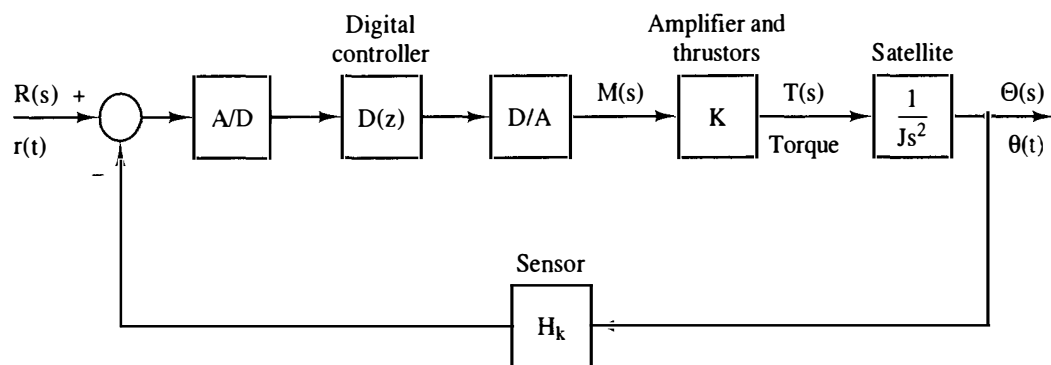


Figure P6-9 Block diagram for a satellite control system.

- 6-10.** It is shown in Example 6.6 that the system of Figure 6-2 has the parameters $\zeta = 0.250$, $\omega_n = 0.9191$, and $\tau = 4.36$ s.
- (a) Find ζ , ω_n , and τ for the sample period $T = 0.5$.
- (b) Repeat part (a) for $T = 0.1$.
- (c) Repeat part (a) for the analog system, that is, the system with the sampler and data hold removed. Note that this case can be considered to be the limit as the sample period approaches zero.

- (d) Give a table listing the three parameters as a function of sampling frequency $f_s = 1/T$. State the result of decreasing the sampling frequency on the parameters.
- 6-11.** Consider the system of Figure P6-11. This system is called a regulator control system, in which it is desired to maintain the output, $c(t)$, at a value of zero in the presence of a disturbance, $f(t)$. In this problem the disturbance is a unit step.
- (a) With $D(z) = 1$ (i.e., no compensation) find the steady-state value of $c(t)$.
- (b) For $f(t)$ to have no effect on the steady-state value of $c(kT)$, $D(z)$ should have a pole at $z = 1$. Let

$$D(z) = 1 + \frac{0.1z}{z - 1}$$

Determine the steady-state value of $c(kT)$.

- (c) Repeat parts (a) and (b) with $T = 1$ s.
- (d) Why does the value of sample period T have no effect on the steady-state response for a constant input?

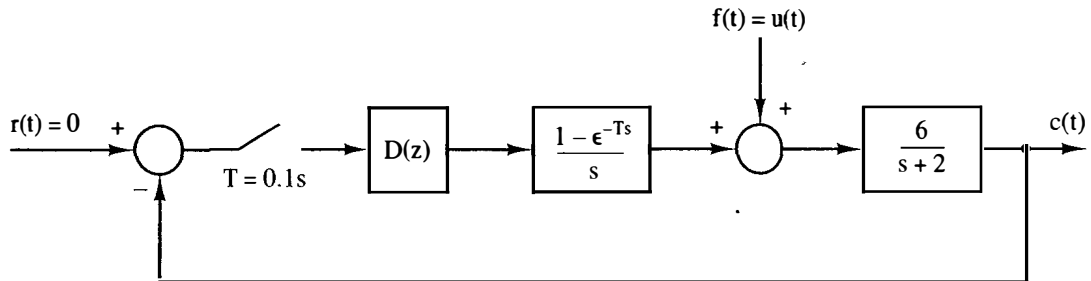


Figure P6-11 System for Problem 6-11.

- 6-12.** Consider the sampled-data systems with the following characteristic equations:
- | | |
|-------------------------------|-------------------------------|
| (i) $z - 0.999 = 0$ | (ii) $z - 0.99 = 0$ |
| (iii) $z - 0.9 = 0$ | (iv) $z + 0.9 = 0$ |
| (v) $z^2 - 1.85z + 0.854 = 0$ | (vi) $z^2 - 1 = 0$ |
| (vii) $z^2 - 2z + 0.99 = 0$ | (viii) $z^2 - 1.2z + 0.7 = 0$ |
- (a) What can you determine about the system natural-response characteristics for each system, for $T = 0.1$ s?
- (b) Repeat part (a) for $T = 1$ s.
- (c) For a given characteristic equation as a function of z , which parameters of the transient response vary with the sample period T , and which are independent of T ?
- 6-13.** Consider the system of Figure P6-1. Suppose that an ideal time delay of 0.2 s is added to the plant, such that the plant transfer function is now given by
- $$G_p(s) = \frac{0.5e^{-0.2T}}{s + 0.5}$$
- (a) Find the time constant of the system if the time delay is omitted.
- (b) Find the time constant of the system if the time delay is included.
- (c) Repeat part (b) for a time delay of 1 s.
- (d) What is the effect on the speed of response of the closed-loop system of adding time delay to the plant?
- 6-14.** (a) Give the system type for the following systems, with $D(z) = 1$. It is not necessary to find the pulse transfer functions to find the system type. Why?

- (i) Figure P6-1.
- (ii) Figure P6-4.
- (iii) Figure P6-7.
- (iv) Figure P6-9.
- (v) Figure P6-11.
- (b) It is desired that the systems of part (a) have zero steady-state error for a constant input. Give the required characteristics for each digital controller.
- (c) It is desired that the systems of part (a) have zero steady-state error for a ramp input. Give the required characteristics for each digital controller.

6-15. Consider the system of Figure P6-15. The digital filter is described by

$$m(kT) = e(kT) - 0.9e[(k-1)T] + m[(k-1)T]$$

- (a) Find the system type.
- (b) Find the steady-state response for a unit-step input, without finding $C(z)$.
- (c) Find the approximate time for the system to reach steady state.
- (d) Find the unit-step response for the system and verify parts (b) and (c).

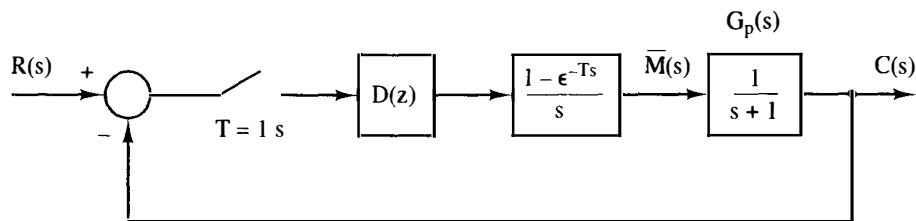


Figure P6-15 System for Problem 6-15.

6-16. Consider the system of Figure P6-15, with the digital filter as described in Problem 6-15.

- (a) Find the system type.
- (b) Find the steady-state response for a unit ramp input, without finding $C(z)$.
- (c) Find the approximate time for the system to reach steady state.
- (d) Calculate the steady-state unit-ramp response for the system to verify parts (b) and (c).

6-17. Consider the system of Problem 6-15. Let the plant transfer function be given by

$$G_p(s) = \frac{s}{s+1}$$

Hence the dc gain of the plant is zero. The digital filter is as given in Problem 6-15.

- (a) Find the system type.
 - (b) Find the steady-state response for a unit-step input, without finding $C(z)$.
 - (c) Find the approximate time for the system to reach steady state.
 - (d) Find $c(kT)$ and verify the results of parts (b) and (c).
- 6-18.** Consider the system of Figure P6-15, with $D(z) = 1$.
- (a) With the sampler and zero-order hold removed, write the system differential equation.
 - (b) Using the rectangular rule for numerical integration with the numerical integration increment equal to 0.25 s, evaluate the unit-step response for $0 \leq t \leq 1.5$ s.
 - (c) Repeat part (b) with the sampler and zero-order hold in the system, and with a sample period $T = 0.5$ s.

- (d) Solve for the exact unit-step responses for parts (b) and (c), and compare results.
- 6-19.** Consider the numerical integration of the differential equation (6-24) in Section 6.6. Applying the rectangular rule results in the difference equation (6-25), which can be expressed as

$$x[(k + 1)H] = x(kH) + H[-x(kH)]$$

- (a) Use the z-transform to solve this equation as a function of $x(0)$ and H .
- (b) Use the results of part (a) to show that as given in Section 6.6, $x(1.0) = 0.3487$ with $x(0) = 1$ and $H = 0.1$.
- (c) Solve for $x(1.0)$ with $x(0) = 1$ and $H = 0.01$.
- (d) The exact value of $x(1.0)$ is given in Section 6.6. Compare the errors in the results of parts (b) and (c).
- 6-20.** Consider the numerical integration of the differential equation

$$\frac{dx(t)}{dt} + x(t) = r(t)$$

using the rectangular rule.

- (a) Develop the difference equation, as in (6-25), for the numerical integration of this differential equation.
- (b) Let $r(t) = 1$, $x(0) = 0$, and $H = 0.1$. Use the z-transform to solve the difference equation of part (a) for $x(1.0)$.
- (c) Repeat part (b) for $H = 0.01$.
- (d) Solve the given differential equation, using the Laplace transform, for the exact value of $x(1.0)$.
- (e) Compare the errors in the results in parts (b) and (c).
- 6-21.** Consider the numerical integration of the differential equation

$$\frac{dx(t)}{dt} + x(t) = 0$$

using the predictor-corrector of Section 6.6. The predictor method is the rectangular rule, and the corrector method is the trapezoidal rule.

- (a) Develop the difference equation, using (6-32) through (6-35), for the numerical integration of the given differential equation. The result should be one difference equation for $x[(k + 1)H]$ as a function of $x(kH)$.
- (b) Let $x(0) = 1$, and $H = 0.1$. Use the z-transform to solve the difference equation of part (a) for $x(1.0)$. This value is given as 0.3685 in Example 6.9.
- (c) Repeat part (b) for $H = 0.33333$. This value is given as 0.3767 in Example 6.9.
- (d) Solve the given differential equation, using the Laplace transform, for the exact value of $x(1.0)$.
- (e) Give the errors in the results in parts (b) and (c).
- 6-22.** Consider the numerical integration of the differential equation

$$\frac{dx(t)}{dt} + x(t) = r(t)$$

using the predictor-corrector of Section 6.6. The predictor method is the rectangular rule, and the corrector method is the trapezoidal rule.

- (a) Develop the difference equation, using (6-32) through (6-35), for the numerical

integration of the given differential equation. The result should be one difference equation for $x(kH)$ as a function of $x[(k-1)H]$ and the input function $r[(k-1)H]$.

- (b) Let $r(t) = 1$, $x(0) = 0$, and $H = 0.1$. Use the z -transform to solve the difference equation of part (a) for $x(1.0)$.
- (c) Repeat part (b) for $H = 0.33333$.
- (d) Solve the given differential equation, using the Laplace transform, for the exact value of $x(1.0)$.
- (e) Give the errors in the results in parts (b) and (c).

6-23. Repeat Problem 6-18, but with the process transfer function given by

$$G(s) = \frac{20}{s(3s + 1)}$$

Solve for the unit-step responses for $0 \leq t \leq 1.0$ s.

Stability Analysis Techniques

7.1 INTRODUCTION

In this chapter stability analysis techniques for linear time-invariant (LTI) discrete-time systems are emphasized. In general, the stability analysis techniques applicable to LTI continuous-time systems may also be applied to the analysis of LTI discrete-time systems, if certain modifications are made. These techniques include the Routh–Hurwitz criterion, root-locus procedures, and frequency-response methods; these techniques are developed in this chapter. The Jury stability test, a technique developed for LTI discrete systems, is also presented.

7.2 STABILITY

To introduce the concepts of stability, consider the LTI system shown in Figure 7-1. For this system,

$$C(z) = \frac{G(z)R(z)}{1 + \overline{GH}(z)} = \frac{K \prod_{i=1}^m (z - z_i)}{\prod_{i=1}^n (z - p_i)} R(z)$$

where z_i are the zeros and p_i the poles of the system transfer function. Using the partial-fraction expansion, for the case that the transfer-function poles are distinct, we can express $C(z)$ as

$$C(z) = \frac{k_1 z}{z - p_1} + \cdots + \frac{k_n z}{z - p_n} + C_R(z) \quad (7-1)$$

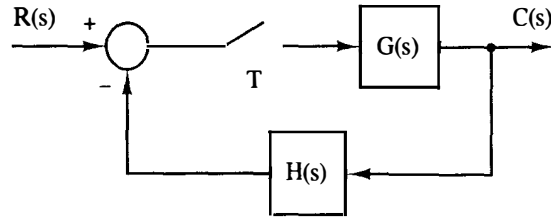


Figure 7-1 Sampled-data system.

where $C_R(z)$ contains the terms of $C(z)$ which originate in the poles of $R(z)$. The first n terms of (7-1) are the natural-response terms of $C(z)$. If the inverse z -transform of these terms tend to zero as time increases, the system is stable, and these terms are called the *transient response*. The inverse z -transform of the i th term is

$$\mathcal{Z}^{-1} \left[\frac{k_i z}{z - p_i} \right] = k_i (p_i)^k \quad (7-2)$$

Thus, if the magnitude of p_i is less than 1, this term approaches zero as k approaches infinity. Note that the factors $(z - p_i)$ originate in the characteristic equation of the system, that is, in

$$1 + \overline{GH}(z) = 0 \quad (7-3)$$

The system is stable provided that all the roots of (7-3) lie inside the unit circle in the z -plane. Of course, (7-3) can also be expressed as

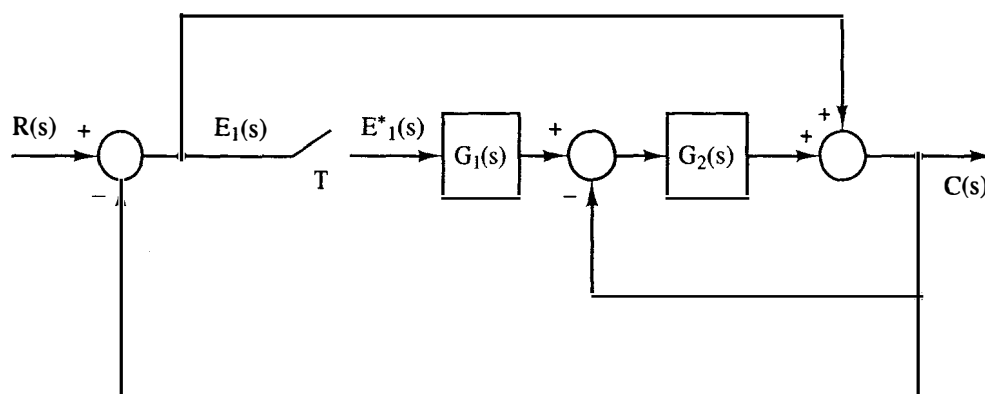
$$1 + \overline{GH}^*(s) = 0 \quad (7-4)$$

and since the area within the unit circle of the z -plane corresponds to the left half of the s -plane, the roots in (7-4) must lie in the left half of the s -plane for stability. The system characteristic equation may be calculated by either (7-3) or (7-4).

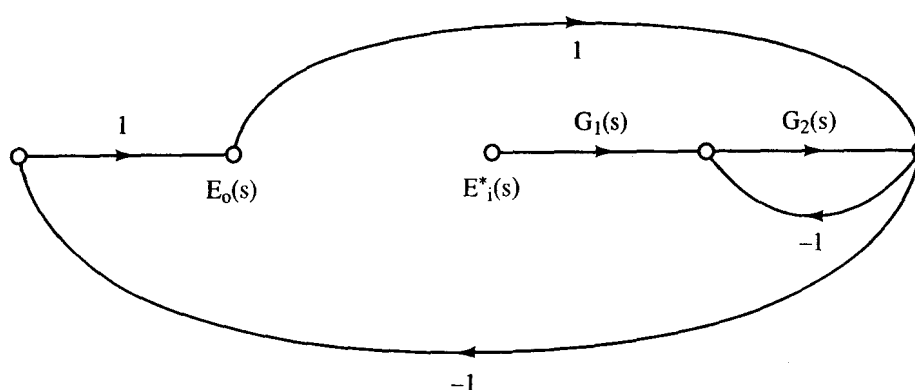
For the case that a root of the characteristic equation is unity in magnitude (e.g., $p_i = 1/\theta$), (7-2) is constant in magnitude. Hence the natural response has a term that neither dies out nor become unbounded as k approaches infinity. If the natural response approaches a bounded nonzero steady state, the system is said to be *marginally stable*. Hence, for a marginally stable system, the characteristic equation (7-3) has at least one zero on the unit circle, with no zeros outside the unit circle.

In the development above it was assumed that the poles of the system transfer function are distinct (no repeated poles). It can be shown that the same condition for stability applies if the transfer function has repeated poles (see Problem 7-1).

We have demonstrated previously that for certain discrete-time control systems, transfer functions cannot be derived. A method for finding the characteristic equation for control systems of this type will now be developed. To illustrate this



(a)



(b)

Figure 7-2 Discrete-time system.

method, consider the system of Figure 7-2a. This system was considered in Example 5.3, and the output expression developed there is

$$C(z) = \left[\frac{R}{2 + G_2} \right](z) + \frac{\left[\frac{G_1 G_2}{2 + G_2} \right](z)}{1 + \left[\frac{G_1 G_2}{2 + G_2} \right](z)} \left[\frac{(1 + G_2)R}{2 + G_2} \right](z)$$

Hence that part of the denominator of $C(z)$ that is independent of the input R is

$$1 + \left[\frac{G_1 G_2}{2 + G_2} \right](z)$$

and this function set equal to zero is then the characteristic equation.

This characteristic function can be developed by a different procedure. Since the stability of a linear system is independent of the input, we set $R(s) = 0$ in the system. In addition, we open the system in front of a sampler and derive a transfer function at this open. We open at a sampler, since we can always write a transfer

function if an input signal is sampled prior to being applied to a continuous-time part of a system. If we opened the system at any other point, we would not be able to find a transfer function. We denote the input signal at this open as $E_i(s)$, and the output signal as $E_o(s)$. Thus, for the system of Figure 7-2a, the system flow graph with no system input and an open at the sampler appears as shown in Figure 7-2b. By Mason's gain formula (or any other applicable technique), we write

$$E_o(s) = \frac{-G_1 G_2}{2 + G_2} E_i^*(s)$$

Taking the z -transform of this equation, we obtain

$$E_o(z) = -\left[\frac{G_1 G_2}{2 + G_2}\right](z) E_i(z)$$

We will denote this open-loop transfer function as

$$G_{op}(z) = \frac{E_o(z)}{E_i(z)} = -\left[\frac{G_1 G_2}{2 + G_2}\right](z)$$

For the closed-loop system, $E_i(z) = E_o(z)$, and the foregoing equations yield

$$[1 - G_{op}(z)]E_o(z) = 0$$

Since we can set initial conditions on the system such that $E_o(z) \neq 0$, then

$$1 - G_{op}(z) = 0$$

and this relationship must be the system characteristic equation. Hence the characteristic equation for this system is

$$1 + \left[\frac{G_1 G_2}{2 + G_2}\right](z) = 0$$

which checks the results of Example 5.3. Another example will now be given.

Example 7.1

Consider the system of Example 5.2, which is repeated in Figure 7-3a. The flow graph of the system is opened at the first sampler as shown in Figure 7-3b. The effect of the second sampler is included, by denoting its input as $E_1(s)$ and its output as $E_1^*(s)$. From this flow graph we write

$$E_1 = G_1 E_i^* - G_2 H E_1^*$$

$$E_o = -G_2 E_1^*$$

Starring the first equation and solving for E_1^* , we obtain

$$E_1^* = \frac{G_1^* E_i^*}{1 + G_2^* H^*}$$

Starring the second equation and substituting in the value for E_1^* , we obtain

$$E_o^* = \frac{-G_1^* G_2^*}{1 + G_2^* H^*} E_i^*$$

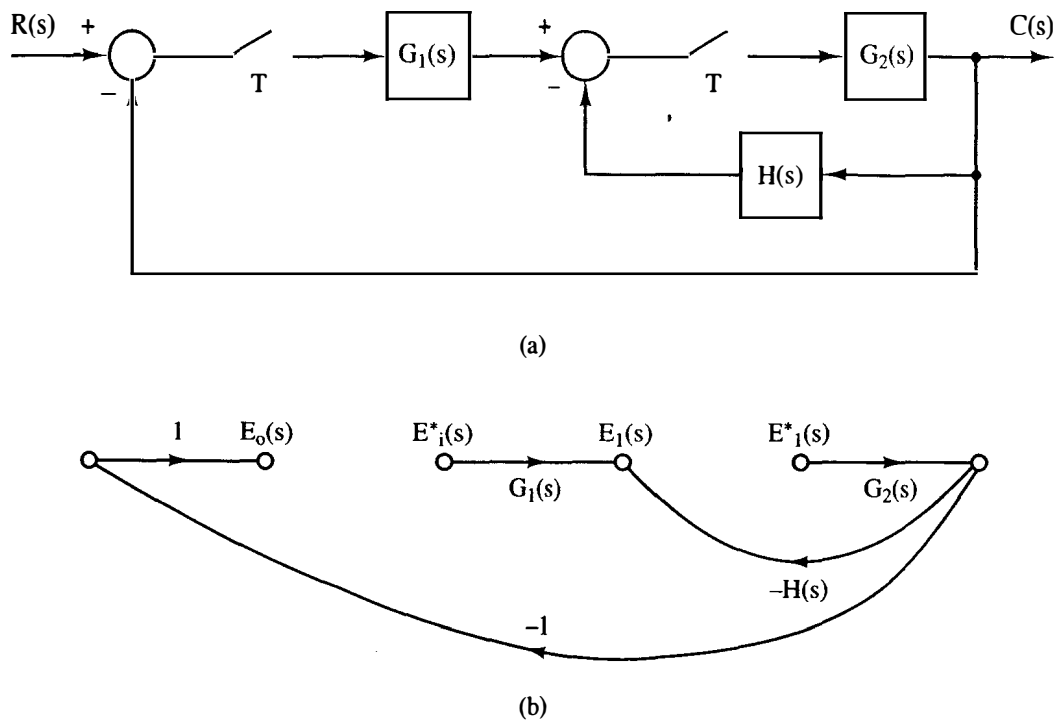


Figure 7-3 System for Example 7.1.

Since $E_i(z) = E_o(z)$ in the closed-loop system,

$$\left[1 + \frac{G_1(z)G_2(z)}{1 + \overline{G_2H}(z)} \right] E_o(z) = 0$$

Thus we can write the characteristic equation

$$1 + G_1(z)G_2(z) + \overline{G_2H}(z) = 0$$

This result is verified in Example 5.2. We leave the derivation of the characteristic equation by opening the system at the second sampler as an exercise for the reader (see Problem 7-2).

From the discussion above, in general, the characteristic equation of a discrete system can be expressed as

$$1 + F(z) = 1 - G_{op}(z) = 0 \quad (7-5)$$

where $G_{op}(z)$ is the *open-loop transfer function*. The function $F(z)$ is important in analysis and design, and we will call it the *open-loop function*. For the system of Figure 7-1, the open-loop function is $\overline{GH}(z)$ and the open-loop transfer function is $-\overline{GH}(z)$.

The characteristic equation of an LTI discrete system can also be calculated from a state-variable approach. Suppose that the state-variable model of the system of Figure 7-1 is

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{r}(k)$$

$$y(k) = \mathbf{C}\mathbf{x}(k) + D\mathbf{r}(k)$$

where the output is now denoted as $y(k)$ rather than $c(k)$. By taking the z -transform of these state equations and eliminating $X(z)$, it was shown in Chapter 2 [see (2-84)] that the system transfer function is given by

$$\frac{Y(z)}{R(z)} = C[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + D$$

The denominator of the transfer function is seen to be $|z\mathbf{I} - \mathbf{A}|$, and thus the characteristic equation for the system is given by

$$|z\mathbf{I} - \mathbf{A}| = 0 \quad (7-6)$$

7.3 BILINEAR TRANSFORMATION

Many analysis and design techniques for continuous-time LTI systems, such as the Routh-Hurwitz criterion and Bode techniques, are based on the property that in the s -plane the stability boundary is the imaginary axis. Thus these techniques cannot be applied to LTI discrete-time systems in the z -plane, since the stability boundary is the unit circle. However, through the use of the transformation

$$z = \frac{1 + (T/2)w}{1 - (T/2)w} \quad (7-7)$$

or solving for w ,

$$w = \frac{2}{T} \frac{z - 1}{z + 1} \quad (7-8)$$

the unit circle of the z -plane transforms into the imaginary axis of the w -plane. This can be seen through the following development. On the unit circle in the z -plane, $z = e^{j\omega T}$ and

$$w = \frac{2}{T} \frac{z - 1}{z + 1} \Big|_{z = e^{j\omega T}} = \frac{2}{T} \frac{e^{j\omega T} - 1}{e^{j\omega T} + 1} = \frac{2}{T} \frac{e^{j\omega T/2} - e^{-j\omega T/2}}{e^{j\omega T/2} + e^{-j\omega T/2}} \quad (7-9)$$

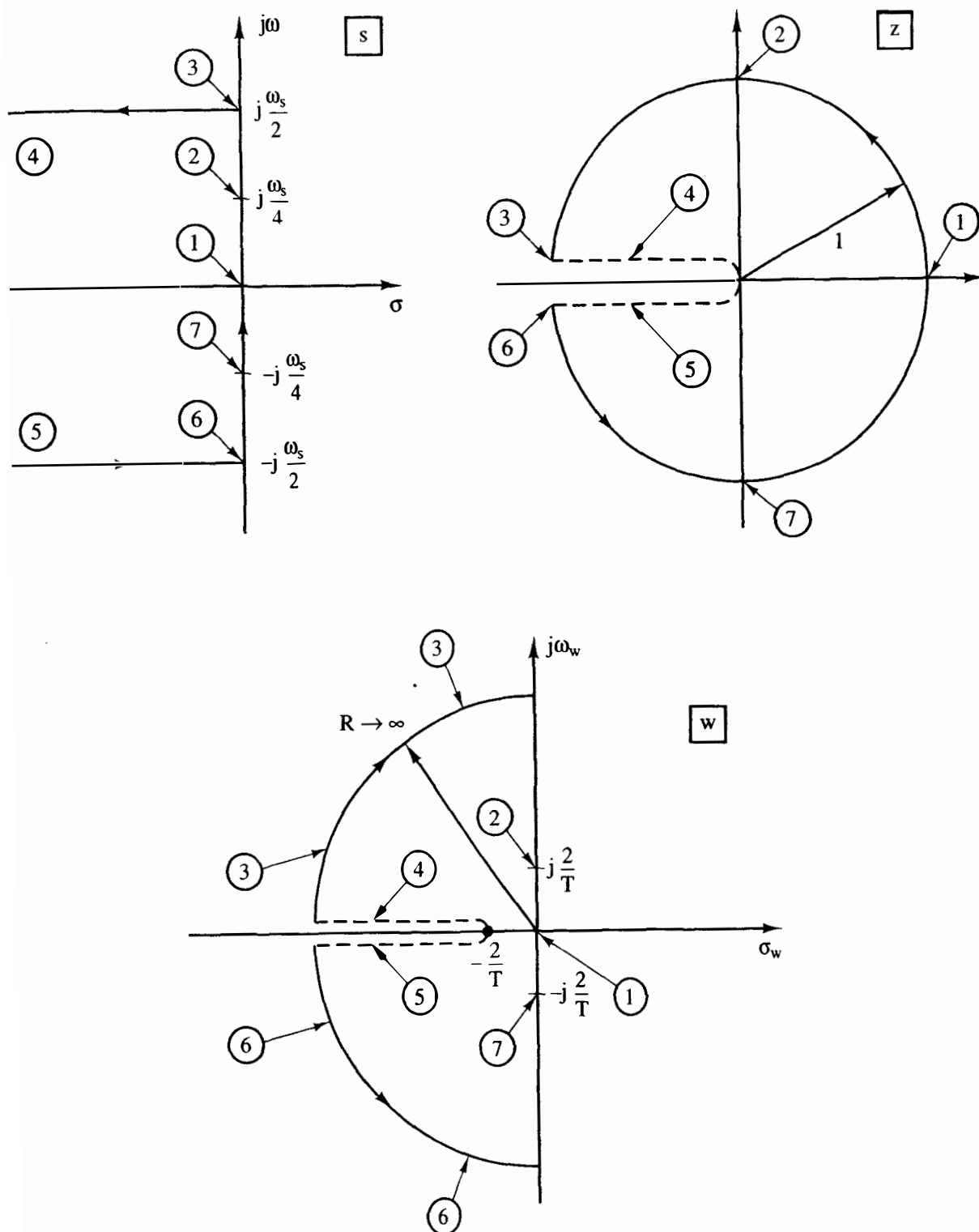
$$w = j \frac{2}{T} \tan \frac{\omega T}{2}$$

Thus it is seen that the unit circle of the z -plane transforms into the imaginary axis of the w -plane. The mappings of the primary strip of the s -plane into both the z -plane ($z = e^{sT}$) and the w -plane are shown in Figure 7-4. It is noted that the stable region of the w -plane is the left half-plane.

Let $j\omega_w$ be the imaginary part of w . We will refer to ω_w as the w -plane frequency. Then (7-9) can be expressed as

$$\omega_w = \frac{2}{T} \tan \frac{\omega T}{2} \quad (7-10)$$

and this expression gives the relationship between frequencies in the s -plane and frequencies in the w -plane.


 Figure 7-4 Mapping from s -plane to z -plane to w -plane.

For small values of real frequency (s -plane frequency) such that ωT is small, (7-10) becomes

$$z \approx 1 + j\omega T$$

Thus the w -plane frequency is approximately equal to the s -plane frequency for this case. The approximation is valid for those values of frequency for which $\tan(\omega T/2) \approx \omega T/2$. For

$$\frac{\omega T}{2} \leq \frac{\pi}{10}, \quad \omega \leq \frac{2\pi}{10T} = \frac{\omega_s}{10} \quad (7-12)$$

the error in this approximation is less than 4 percent. Because of the phase lag introduced by the zero-order hold, we usually choose the sample period T such that (7-12) is satisfied over most if not all of the band of frequencies that the system will pass (the system bandwidth). At $\omega = \omega_s/10$, the zero-order hold introduces a phase lag of 18° (see Figure 3-13), which is an appreciable amount and, as we shall see, can greatly affect system stability.

7.4 THE ROUTH–HURWITZ CRITERION

The Routh–Hurwitz criterion [1] may be used in the analysis of LTI continuous-time systems to determine if any roots of a given equation are in the right half of the s -plane. If this criterion is applied to the characteristic equation of an LTI discrete-time system when expressed as a function of z , no useful information on stability is obtained. However, if the characteristic equation is expressed as a function of the bilinear transform variable w , then the stability of the system may be determined by directly applying the Routh–Hurwitz criterion.

We assume that the reader is familiar with the procedures for applying the Routh–Hurwitz criterion. The procedure is summarized in Table 7-1. The technique will now be illustrated via examples.

Example 7.2

Consider the system shown in Figure 7-5, with $T = 0.1$ s. The open-loop function is

$$G(s) = \frac{1 - e^{-Ts}}{s} \left[\frac{1}{s(s+1)} \right]$$

Hence, from the z -transform tables we obtain

$$\begin{aligned} G(z) &= \frac{z-1}{z} \left[\frac{(\epsilon^{-T} + T - 1)z^2 + (1 - \epsilon^{-T} - T\epsilon^{-T})z}{(z-1)^2(z - \epsilon^{-T})} \right] \\ &= \frac{0.00484z + 0.00468}{(z-1)(z - 0.905)} \end{aligned}$$

Then $G(w)$ is given by

$$G(w) = G(z)|_{z = [1 + (T/2)w]/[1 - (T/2)w]} = G(z)|_{z = (1 + 0.05w)/(1 - 0.05w)}$$

or

$$G(w) = \frac{-0.00016w^2 - 0.1872w + 3.81}{3.81w^2 + 3.80w}$$

TABLE 7-1 BASIC PROCEDURE FOR APPLYING THE ROUTH–HURWITZ CRITERION

1. Given a characteristic equation of the form

$$F(w) = b_n w^n + b_{n-1} w^{n-1} + \cdots + b_1 w + b_0 = 0$$

form the Routh array as

w^n	b_n	b_{n-2}	b_{n-4}	\cdots
w^{n-1}	b_{n-1}	b_{n-3}	b_{n-5}	\cdots
w^{n-2}	c_1	c_2	c_3	\cdots
\vdots	d_1	d_2	d_3	\cdots
w^1	j_1			
w^0	k_1			

2. Only the first two rows of the array are obtained from the characteristic equation. The remaining rows are calculated as follows.

$$c_1 = \frac{b_{n-1}b_{n-2} - b_n b_{n-3}}{b_{n-1}} \quad d_1 = \frac{c_1 b_{n-3} - b_{n-1} c_2}{c_1}$$

$$c_2 = \frac{b_{n-1}b_{n-4} - b_n b_{n-5}}{b_{n-1}} \quad d_2 = \frac{c_1 b_{n-5} - b_{n-1} c_3}{c_1}$$

$$c_3 = \frac{b_{n-1}b_{n-6} - b_n b_{n-7}}{b_{n-1}} \quad \vdots$$

3. Once the array has been formed, the Routh–Hurwitz criterion states that the number of roots of the characteristic equation with positive real parts is equal to the number of sign changes of the coefficients in the first column of the array.
4. Suppose that the w^{i-1} th row contains only zeros, and that the w^i th row directly above it has the coefficients $\alpha_1, \alpha_2, \dots$. The auxiliary equation is then

$$\alpha_1 w^i + \alpha_2 w^{i-2} + \alpha_3 w^{i-4} + \cdots = 0$$

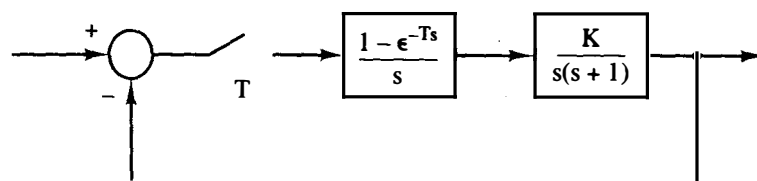
This equation is a factor of the characteristic equation.

Then the characteristic equation is given by

$$1 + KG(w) = (3.81 - 0.00016K)w^2 + (3.80 - 0.1872K)w + 3.81K = 0$$

The Routh array derived from this equation is

w^2	$3.81 - 0.00016K$	$3.81K$	$\Rightarrow K < 23,813$
w^1	$3.80 - 0.1872K$		$\Rightarrow K < 20.3$
w^0	$3.81K$		$\Rightarrow K > 0$


Figure 7-5 System for Examples 7.2 and 7.3.

Hence, for no sign changes to occur in the first column, it is necessary that K be in the range $0 < K < 20.3$, and this is the range of K for stability. The computer programs described in Appendix VI will calculate w -plane transfer functions.

Example 7.3

Consider again the system of Example 6.4 (shown again in Figure 7-5 with $T = 1$ s) with a gain factor K added to the plant. The characteristic equation is given by

$$1 + KG(w) = 1 + KG(z)|_{z = (1 + 0.5w)/(1 - 0.5w)}$$

$$= 1 + \frac{K \left[0.368 \left[\frac{1 + 0.5w}{1 - 0.5w} \right] + 0.264 \right]}{\left[\frac{1 + 0.5w}{1 - 0.5w} \right]^2 - 1.368 \left[\frac{1 + 0.5w}{1 - 0.5w} \right] + 0.368}$$

or

$$1 + KG(w) = 1 + \frac{-0.0381K(w - 2)(w + 12.14)}{w(w + 0.924)}$$

$$= \frac{(1 - 0.0381K)w^2 + (0.924 - 0.386K)w + 0.924K}{w(w + 0.924)}$$

Thus the characteristic equation may be expressed as

$$(1 - 0.0381K)w^2 + (0.924 - 0.386K)w + 0.924K = 0$$

The Routh array is then

w^2	$1 - 0.0381K$	$0.924K$	\Rightarrow	$K < 26.2$
w^1	$0.924 - 0.386K$		\Rightarrow	$K < 2.39$
w^0	$0.924K$		\Rightarrow	$K > 0$

Hence the system is stable for $0 < K < 2.39$.

From our knowledge of continuous-time systems, we know that the Routh–Hurwitz criterion can be used to determine the value of K at which the root locus crosses into the right half-plane (i.e., the value of K at which the system becomes unstable). That value of K is the gain at which the system is *marginally stable*, and thus can also be used to determine the resultant frequency of steady-state oscillation. Therefore, $K = 2.39$ in Example 7.3 is the gain for which the system is marginally stable.

In a manner similar to that employed in continuous-time systems, the frequency of oscillation at $K = 2.39$ can be found from the w^2 row of the array. Recalling that ω_w is the imaginary part of w , we obtain the auxiliary equation (see Table 7-1)

$$(1 - 0.0381K)w^2 + 0.924K|_{K=2.39} = 0.9089w^2 + 2.181 = 0$$

or

$$w = \pm j \sqrt{\frac{2.181}{0.9089}} = \pm j1.549$$

Then $\omega_w = 1.549$ and from (7-10),

$$\omega = \frac{2}{T} \tan^{-1} \frac{\omega_w T}{2} = \frac{2}{1} \tan^{-1} \left[\frac{(1.549)(1)}{2} \right] = 1.32 \text{ rad/s}$$

and is the s -plane (real) frequency at which this system will oscillate with $K = 2.39$.

The same system was used in both examples in this section, but with different sample periods. For $T = 0.1$ s, the system is stable for $0 < K < 20.3$. For $T = 1$ s, the system is stable for $0 < K < 2.39$. Hence we can see the dependency of system stability on the sample period. The degradation of stability with increasing T (decreasing sampling frequency) is due to the delay (phase lag) introduced by the sampler and data hold, which is illustrated in the frequency response of the data hold, shown in Figure 3-13.

7.5 JURY'S STABILITY TEST

For continuous-time systems, the Routh–Hurwitz criterion offers a simple and convenient technique for determining the stability of low-ordered systems. However, since the stability boundary in the z -plane is different from that in the s -plane, the Routh–Hurwitz criterion cannot be directly applied to discrete-time systems if the system characteristic equation is expressed as a function of z . A stability criterion for discrete-time systems that is similar to the Routh–Hurwitz criterion and can be applied to the characteristic equation written as a function of z is the Jury stability test [2].

Jury's test will now be presented. Let the characteristic equation of a discrete-time system be expressed as

$$Q(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 = 0, \quad a_n > 0 \quad (7-13)$$

Then form the array as shown in Table 7-2. Note that the elements of each of the

TABLE 7-2 ARRAY FOR JURY'S STABILITY TEST

z^0	z^1	z^2	\dots	z^{n-k}	\dots	z^{n-1}	z^n
a_0	a_1	a_2	\dots	a_{n-k}	\dots	a_{n-1}	a_n
a_n	a_{n-1}	a_{n-2}	\dots	a_k	\dots	a_1	a_0
b_0	b_1	b_2	\dots	b_{n-k}	\dots	b_{n-1}	
b_{n-1}	b_{n-2}	b_{n-3}	\dots	b_{k-1}	\dots	b_0	
c_0	c_1	c_2	\dots	c_{n-k}	\dots		
c_{n-2}	c_{n-3}	c_{n-4}	\dots	c_{k-2}	\dots		
\vdots	\vdots	\vdots	\vdots	\vdots			
l_0	l_1	l_2	l_3				
l_3	l_2	l_1	l_0				
m_0	m_1	m_2					

even-numbered rows are the elements of the preceding row in reverse order. The elements of the odd-numbered rows are defined as

$$\begin{aligned} b_k &= \begin{vmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{vmatrix}, & c_k &= \begin{vmatrix} b_0 & b_{n-1-k} \\ b_{n-1} & b_k \end{vmatrix} \\ d_k &= \begin{vmatrix} c_0 & c_{n-2-k} \\ c_{n-2} & c_k \end{vmatrix} \dots \end{aligned} \quad (7-14)$$

The necessary and sufficient conditions for the polynomial $Q(z)$ to have no roots outside or on the unit circle, with $a_n > 0$, are as follows:

$$\begin{aligned} Q(1) &> 0 \\ (-1)^n Q(-1) &> 0 \\ |a_0| &< a_n \\ |b_0| &> |b_{n-1}| \\ |c_0| &> |c_{n-2}| \\ |d_0| &> |d_{n-3}| \\ &\vdots \\ |m_0| &> |m_2| \end{aligned} \quad (7-15)$$

Note that for a second-order system, the array contains only one row. For each additional order, two additional rows are added to the array. Note also that for an n th-order system, there are a total of $n + 1$ constraints.

Jury's test may be applied in the following manner:

1. Check the three conditions $Q(1) > 0$, $(-1)^n Q(-1) > 0$, and $|a_0| < a_n$, which requires no calculations. Stop if any of these conditions are not satisfied.
2. Construct the array, checking the conditions of (7-15) as each row is calculated. Stop if any condition is not satisfied.

Example 7.4

Consider again the system of Example 6.4 (and Example 7.3). Suppose that a gain factor K is added to the plant, and it is desired to determine the range of K for which the system is stable. Now, from Example 6.4, the system characteristic equation is

$$1 + KG(z) = 1 + \frac{(0.368z + 0.264)K}{z^2 - 1.368z + 0.368} = 0$$

or

$$z^2 + (0.368K - 1.368)z + (0.368 + 0.264K) = 0$$

The Jury array is

z^0	z^1	z^2
$0.368 + 0.264K$	$0.368K - 1.368$	1

The constraint $Q(1) > 0$ yields

$$1 + (0.368K - 1.368) + (0.368 + 0.264K) = 0.632K > 0 \Rightarrow K > 0$$

The constraint $(-1)^2 Q(-1) > 0$ yields

$$1 - 0.368K + 1.368 + 0.368 + 0.264K > 0 \Rightarrow K < \frac{2.736}{0.104} = 26.3$$

The constraint $|a_0| < a_2$ yields

$$0.368 + 0.264K < 1 \Rightarrow K < \frac{0.632}{0.264} = 2.39$$

Thus the system is stable for

$$0 < K < 2.39$$

The system is marginally stable for $K = 2.39$. For this value of K , the characteristic equation is

$$z^2 + (0.368K - 1.368)z + (0.368 + 0.264K)|_{K=2.39} = z^2 - 0.488z + 1 = 0$$

The roots of this equation are

$$z = 0.244 \pm j0.970 = 1/\pm 75.9^\circ = 1/\pm 1.32 \text{ rad} = 1/\pm \omega T$$

Since $T = 1$ s, the system will oscillate at a frequency of 1.32 rad/s. These results verify those of Example 7.3.

Example 7.5

Suppose that the characteristic equation for a closed-loop discrete-time system is given by the expression

$$Q(z) = z^3 - 1.8z^2 + 1.05z - 0.20 = 0$$

The first conditions of Jury's test are

$$Q(1) = 1 - 1.8 + 1.05 - 0.2 = 0.05 > 0$$

$$(-1)^3 Q(-1) = -[-1 - 1.8 - 1.05 - 0.2] = 4.05 > 0$$

$$|a_0| = 0.2 < a_3 = 1$$

The Jury array is calculated to be

z^0	z^1	z^2	z^3
-0.2	1.05	-1.8	1
1	-1.8	1.05	-0.2
-0.96	1.59	-0.69	

where the last row has been calculated as follows:

$$b_0 = \begin{vmatrix} -0.2 & 1 \\ 1 & -0.2 \end{vmatrix} = -0.96, \quad b_1 = \begin{vmatrix} -0.2 & -1.8 \\ 1 & 1.05 \end{vmatrix} = 1.59$$

$$b_2 = \begin{vmatrix} -0.2 & 1.05 \\ 1 & -1.8 \end{vmatrix} = -0.69$$

Hence the last condition is

$$|b_0| = 0.96 > |b_2| = 0.69$$

Therefore, since all conditions are satisfied, the system is stable. The characteristic equation can be factored as

$$Q(z) = (z - 0.5)^2(z - 0.8)$$

This form of the equation clearly indicates the system's stability.

Example 7.6

In this example we wish to determine the range of values of K_p in Example 6.8 so that the system is stable. Recall that in Example 6.8, a proportional-plus-integral (PI) compensator was designed to satisfy certain steady-state constraints. In this example we derive the requirements for this compensated system to be stable. The characteristic equation for the system is given by the expression

$$1 + D(z)G(z) = 0$$

Or, from Example 6.8,

$$1 + \frac{(K_I + K_p)z - K_p}{z - 1} \left[\frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}} \right] = 0$$

Thus

$$z^2 - [(1 + \epsilon^{-T}) - (1 - \epsilon^{-T})(K_I + K_p)]z + \epsilon^{-T} - (1 - \epsilon^{-T})K_p = 0$$

Suppose that $T = 0.1$ s and $K_I = 100T = 10$, as required in Example 6.8. Then the characteristic equation becomes

$$z^2 - (0.953 - 0.0952K_p)z + 0.905 - 0.0952K_p = 0$$

For this system, the Jury array is

z^0	z^1	z^2
$0.905 - 0.0952K_p$	$0.0952K_p - 0.953$	1

The constraint $Q(1) > 0$ yields

$$1 + 0.0952K_p - 0.953 + 0.905 - 0.0952K_p > 0$$

Thus this constraint is satisfied independent of K_p . The constraint $(-1)^2 Q(-1) > 0$ yields

$$1 - 0.0952K_p + 0.953 + 0.905 - 0.0952K_p > 0$$

or since K_p is normally positive,

$$0 < K_p < 15.01$$

The constraint $|a_0| < a_2$ yields

$$|0.905 - 0.0952K_p| < 1$$

or

$$K_p < 20.0$$

Thus the stability constraint on positive K_p is $K_p < 15.01$, in order that the steady-state error constraint in Example 6.8 be satisfied.

7.6 ROOT LOCUS

For the LTI sampled-data system of Figure 7-6,

$$\frac{C(z)}{R(z)} = \frac{KG(z)}{1 + KGH(z)}$$

The system characteristic equation is, then,

$$1 + KGH(z) = 0 \quad (7-16)$$

The root locus for this system is a plot of the locus of roots in (7-16) in the z -plane as a function of K . Thus the rules of root-locus construction for discrete-time systems are identical to those for continuous-time systems, since the roots of any equation are dependent only on the coefficients of the equation and are independent of the designation of the variable. Since the rules for root-locus construction are numerous and appear in any standard text for continuous-time control systems [1,4], only the most important rules will be repeated here in abbreviated form. These rules are given in Table 7-3.

While the rules for construction of both s -plane and z -plane root loci are the same, there are important differences in the interpretation of the root loci. For example, in the z -plane, the stable region is the interior of the unit circle. In addition, root locations in the z -plane have different meanings from those in the s -plane from the standpoint of the system time response, as was seen in Figure 6-11.

The two examples that follow review not only the root-locus technique, but also the part it plays in stability analyses.

Example 7.7



Consider again the system of Example 6.4. For this system

$$KG(z) = \frac{0.368K(z + 0.717)}{(z - 1)(z - 0.368)}$$

Thus the loci originate at $z = 1$ and $z = 0.368$, and terminate at $z = -0.717$ and $z = \infty$. There is one asymptote, at 180° . The breakaway points, obtained from

$$\frac{d}{dz}[G(z)] = 0$$

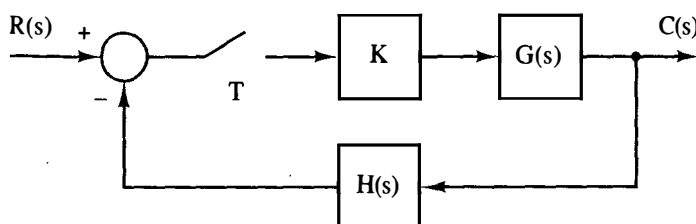


Figure 7-6 Sampled-data system.

TABLE 7-3 RULES FOR ROOT-LOCUS CONSTRUCTION

For the characteristic equation

$$1 + K\overline{GH}(z) = 0$$

1. Loci originate on poles of $\overline{GH}(z)$ and terminate on the zeros of $\overline{GH}(z)$.
2. The root locus on the real axis always lies in a section of the real axis to the left of an odd number of poles and zeros on the real axis.
3. The root locus is symmetrical with respect to the real axis.
4. The number of asymptotes is equal to the number of poles of $\overline{GH}(z)$, n_p , minus the number of zeros of $\overline{GH}(z)$, n_z , with angles given by $(2k + 1)\pi/(n_p - n_z)$.
5. The asymptotes intersect the real axis at σ , where

$$\sigma = \frac{\sum \text{poles of } \overline{GH}(z) - \sum \text{zeros of } \overline{GH}(z)}{n_p - n_z}$$

6. The breakaway points are given by the roots of

$$\frac{d[\overline{GH}(z)]}{dz} = 0$$

or, equivalently,

$$D(z) \frac{dN(z)}{dz} - N(z) \frac{dD(z)}{dz} = 0, \quad \overline{GH}(z) = \frac{N(z)}{D(z)}$$

occur at $z = 0.65$ for $K = 0.196$, and $z = -2.08$ for $K = 15.0$. The root locus is then as shown in Figure 7-7. The points of intersection of the root loci with the unit circle may be found by graphical construction, the Jury stability test, or the Routh-Hurwitz criterion. To illustrate the use of the Jury stability test, consider the results of Example 7.4. The value of gain for marginal stability (i.e., for the roots to appear on the unit circle) is $K = 2.39$. For this value of gain, the characteristic equation is, from Example 7.4,

$$z^2 - 0.488z + 1 = 0$$

The roots of this equation are

$$z = 0.244 \pm j0.970 = 1/\pm 75.8^\circ = 1/\pm 1.32 \text{ rad} = 1/\pm \omega T$$

and thus these are the points at which the root locus crosses the unit circle. Note that the frequency of oscillation for this case is $\omega = 1.32$, since $T = 1$ s. This was also calculated in Example 7.3 using the Routh-Hurwitz criterion, and in Example 7.4 using the Jury test.

The value of the gain at points where the root locus crosses the unit circle can also be determined using the root-locus condition that at any point along the locus the magnitude of the open-loop function must be equal to 1 [i.e., $|K\overline{GH}(z)| = 1$]. Using the condition and Figure 7-8, we note that

$$\frac{0.368K(Z_1)}{(P_1)(P_2)} = 1$$

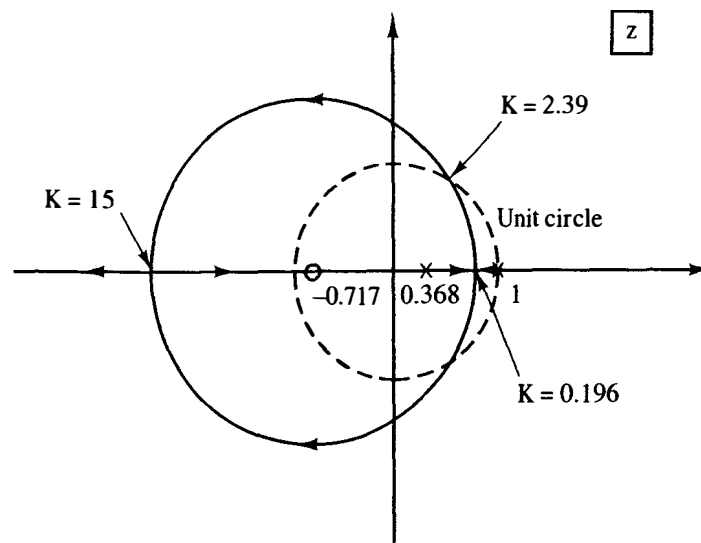


Figure 7-7 Root locus for Example 7.7.

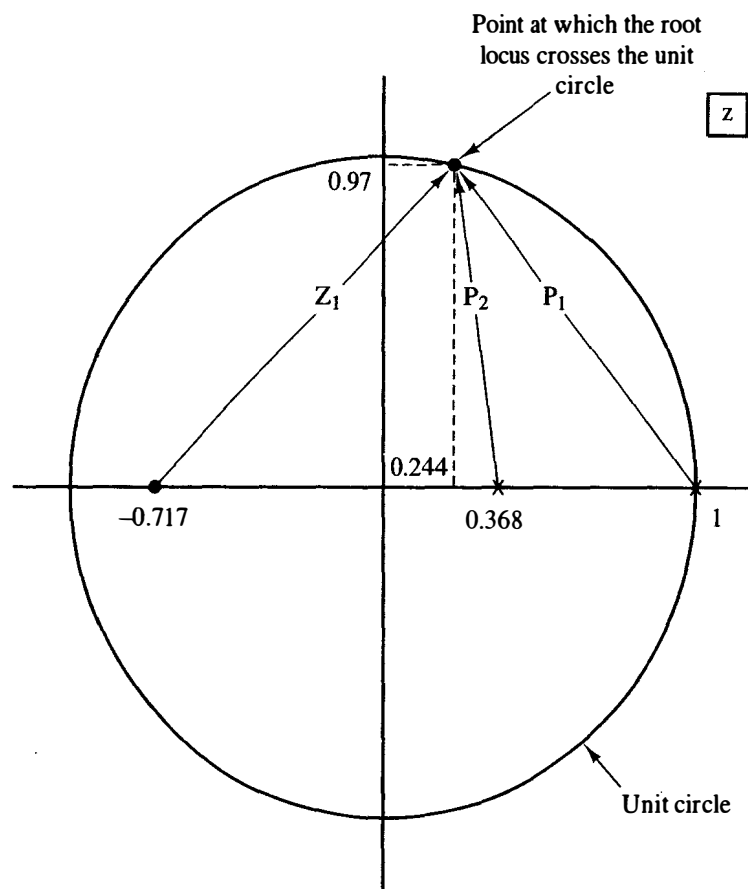


Figure 7-8 Determination of the system gain at the crossover point on the unit circle.

From Figure 7-8 the following values can be calculated: $Z_1 = 1.364$, $P_1 = 1.229$, and $P_2 = 0.978$. Using these values in the equation above yields $K = 2.39$. A MATLAB program that solves for and plots a root locus for this example, for $K = 0, 0.1, 0.2, \dots, 1.0$, is given by

```
format compact
k = 0:0.1:1;
n = [0 0.368 0.264];
d = [1 -1.368 0.368];
r = rlocus(n,d,k);
k
r
pause
plot(real(r),imag(r), 'x')
title('Root Locus')
```

Example 7.8

Consider the system shown in Figure 7-9. The transfer function $[1 + 10s]$ is that of an analog proportional-plus-derivative (PD) controller [1]. The digital PD controller is covered in Chapter 8. We wish to determine the form of the root locus and the range of K for stability.

We denote the open-loop function as

$$KG(s) = \frac{1 - e^{-sT}}{s} \left[\frac{K(1 + 10s)}{s^2} \right]$$

Applying the z -transform, we obtain

$$KG(z) = \frac{10.5K(z - 0.9048)}{(z - 1)^2}$$

The loci originate at $z = 1$ and terminate at $z = 0.9048$ and $z = -\infty$. There is one asymptote at 180° . The root locus is shown in Figure 7-10. The system becomes unstable when the closed-loop pole leaves the interior of the unit circle at point A shown in Figure 7-10. The value of K at this point can be determined from the condition $KG(z) = -1$. Therefore,

$$\left. \frac{10.5K(z - 0.9048)}{(z - 1)^2} \right|_{z=-1} = \frac{10.5K(-1.9048)}{4} = -1$$

Thus $K = 0.2$, and we see that the system is stable of $0 < K < 0.2$.

7.7 THE NYQUIST CRITERION

To develop the Nyquist criterion for discrete-time systems, we consider the two systems shown in Figure 7-11. In Figure 7-11b, $G(s)$ contains the transfer function of a data hold. The transfer function for the continuous-time system of Figure 7-11a is

$$\frac{C(s)}{R(s)} = \frac{G_p(s)}{1 + G_p(s)H(s)} \quad (7-17)$$

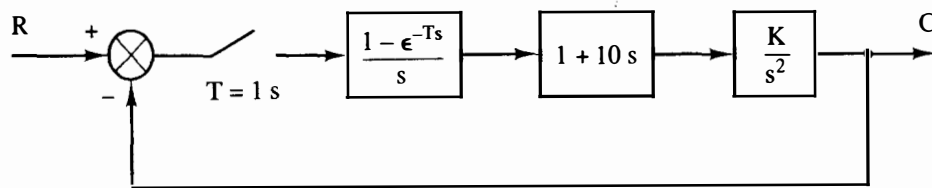


Figure 7-9 System for Example 7.8.

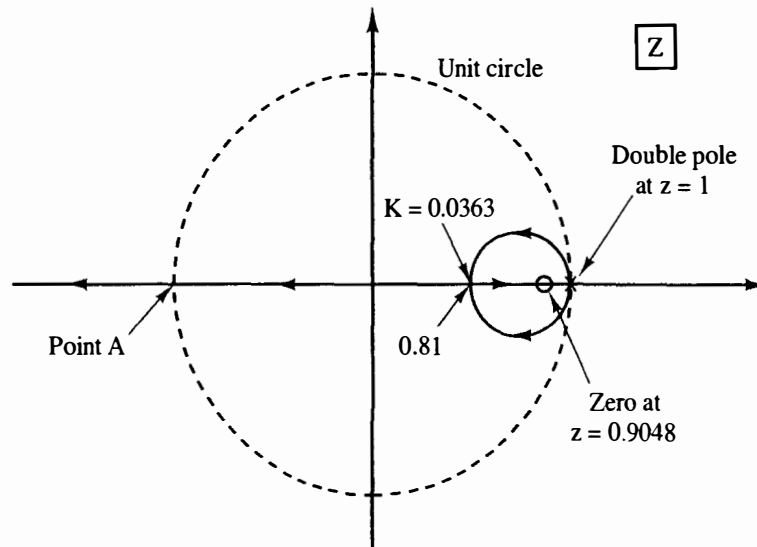


Figure 7-10 Root locus for the system of Example 7.8.

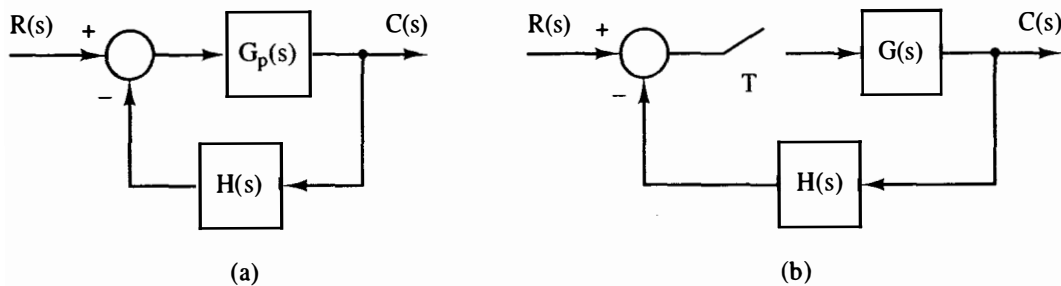


Figure 7-11 Continuous-time and sampled-data systems.

and for the sampled-data system of Figure 7-11b the transfer function is

$$\frac{C^*(s)}{R^*(s)} = \frac{G^*(s)}{1 + \overline{GH}^*(s)} \quad (7-18)$$

Thus the characteristic equation for the continuous-time system is

$$1 + G_p(s)H(s) = 0 \quad (7-19)$$

and for the sampled-data system, the characteristic equation is

$$1 + \overline{GH}^*(s) = 0 \quad (7-20)$$

The characteristic equation of the sampled-data system can also be written as

$$1 + \overline{GH}(z) = 0 \quad (7-21)$$

Recall that the continuous-time system is stable if the roots of (7-19) are all contained in the left half-plane. Similarly, the sampled-data system is stable if the roots of (7-20) all lie in the left half-plane, or if the roots of (7-21) all lie within the unit circle.

The Nyquist criterion is based on Cauchy's principle of argument [5].

Theorem. Let $f(z)$ be the ratio of two polynomials in z . Let the closed curve C in the z -plane be mapped into the complex plane through the mapping $f(z)$. If $f(z)$ is analytic within and on C , except at a finite number of poles, and if $f(z)$ has neither poles nor zeros on C , then

$$N = Z - P$$

where Z is the number of zeros of $f(z)$ in C , P is the number of poles of $f(z)$ in C , and N is the number of encirclements of the origin, taken in the same sense as C .

In order to determine the stability of an analog system via the Nyquist criterion, a complex-plane plot of the open-loop function, $G_p(s)H(s)$, is made. This plot is referred to as the Nyquist diagram. The closed curve of C in Cauchy's principle of argument is called the Nyquist path, and for continuous-time systems encloses the right half-plane as shown in Figure 7-12. The Nyquist criterion states that if the complex-plane plot of $G_p(s)H(s)$, for values of s on the Nyquist path, is made, then

$$N = Z - P \Rightarrow Z = N + P \quad (7-22)$$

where N is the number of clockwise encirclements of the -1 point made by the plot of $G_p(s)H(s)$, Z is the number of zeros of the characteristic equation enclosed by the Nyquist path, and P is the number of poles of the open-loop function enclosed by the Nyquist path. Since the Nyquist path encloses the right half-plane, Z in (7-22) must be zero for stability.

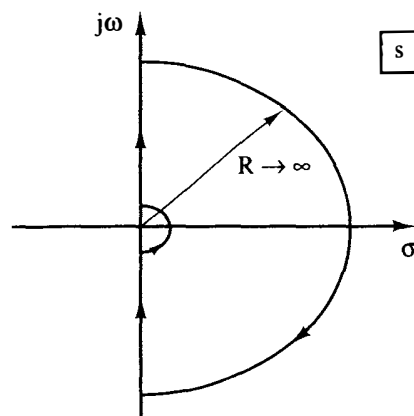


Figure 7-12 Nyquist path in the s -plane

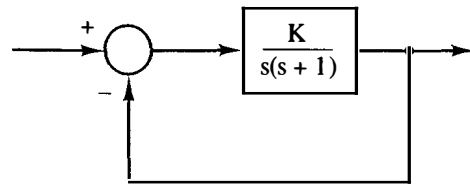


Figure 7-13 System for Example 7.9.

Example 7.9

As an example of the Nyquist criterion applied to a continuous-time system, consider the system of Figure 7-13. The Nyquist path and the resultant Nyquist diagram are shown in Figure 7-14. The small detour taken by the Nyquist path around the origin, labeled I on Figure 7-14a, is necessary, since $G_p(s)$ has a pole at the origin. Along this detour, let

$$s = \rho e^{j\theta}, \quad \rho \ll 1$$

Then

$$G_p(s)|_{s=\rho e^{j\theta}} \approx \frac{K}{\rho e^{j\theta}} = \frac{K}{\rho} \angle -\theta$$

Thus the small detour generates the large arc on the Nyquist diagram. For the Nyquist path along the $j\omega$ -axis (II),

$$G_p(s)|_{s=j\omega} = \frac{K}{j\omega(1+j\omega)} = \frac{K}{\omega\sqrt{1+\omega^2}} \angle -90^\circ - \tan^{-1}\omega$$

This portion of the Nyquist diagram is simply the frequency response of the system. Along the large arc of the Nyquist path, $G_p(s) \approx 0$. The complete Nyquist diagram is given in Figure 7-14b. Now let us apply the Nyquist criterion. We can see from the Nyquist diagram in Figure 7-14 that $N = 0$. Since the open-loop transfer function has no poles in the right half-plane, $P = 0$, and therefore from the equation

$$Z = N + P$$

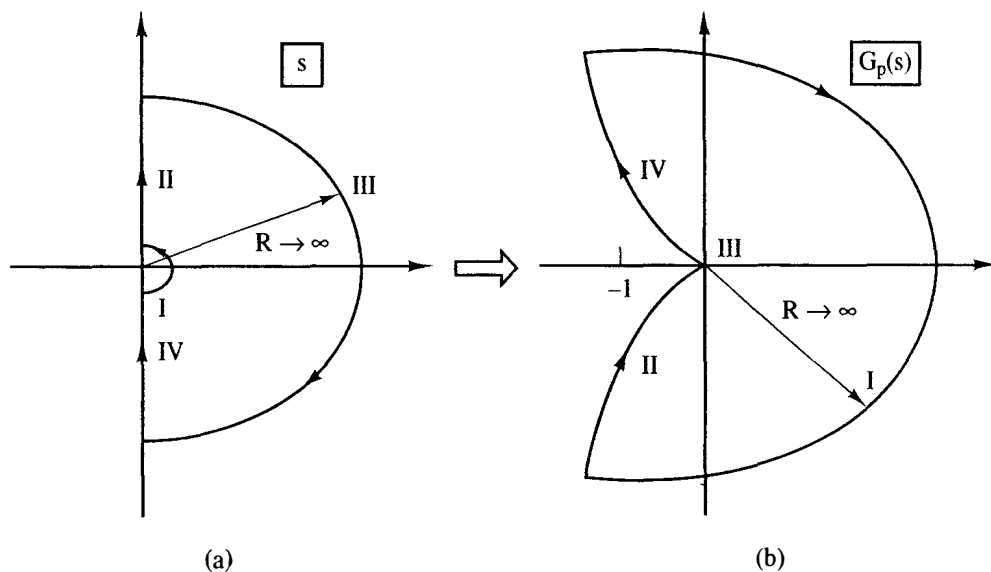


Figure 7-14 Nyquist diagram for Example 7.9.

we find that $Z = 0$. Thus the system is stable. In addition, from the Nyquist diagram in Figure 7-14 it can be seen that the system is stable for all $K > 0$.

Consider now the sampled-data system of Figure 7-11b. The characteristic equation is given in (7-20), and thus the Nyquist diagram for the system may be generated by using the s -plane Nyquist path of Figure 7-12 (i.e., the same path as that used for continuous-time systems). However, recall from Chapter 3 that $\overline{GH}^*(s)$ is periodic in s with period $j\omega_s$. Thus it is necessary only that $\overline{GH}^*(j\omega)$ be plotted for $-\omega_s/2 \leq \omega \leq \omega_s/2$, in order to obtain the frequency response. Consider also the relationship, from (3-11),

$$\begin{aligned}\overline{GH}^*(j\omega) &= \frac{1}{T} \sum_{n=-\infty}^{\infty} GH(j\omega + jn\omega_s) \\ &= \frac{1}{T} [GH(j\omega) + GH(j\omega + j\omega_s) + GH(j\omega - j\omega_s) + \cdots]\end{aligned}\quad (7-23)$$

Since physical systems are generally low pass, $\overline{GH}^*(j\omega)$ may be approximated by only a few terms of (7-23). In general, however, the approximation will not apply for $\omega > \omega_s/2$. A digital computer program may be written for (7-23), and thus the Nyquist diagram of a sampled-data system may be obtained without calculating the z -form of the transfer function.

The Nyquist diagram may also be generated directly from the z -plane. The Nyquist path for the z -plane is the unit circle, and the path direction is counterclockwise, as shown in Figure 7-15. To apply Cauchy's principle of argument, let Z_i and P_i be the zeros of the characteristic equation and the poles of the open-loop function, respectively, inside the unit circle; and let Z_o and P_o be the zeros of the characteristic equation and the poles of the open-loop function, respectively, outside the unit circle. Then, from (7-22),

$$N = -(Z_i - P_i) \quad (7-24)$$

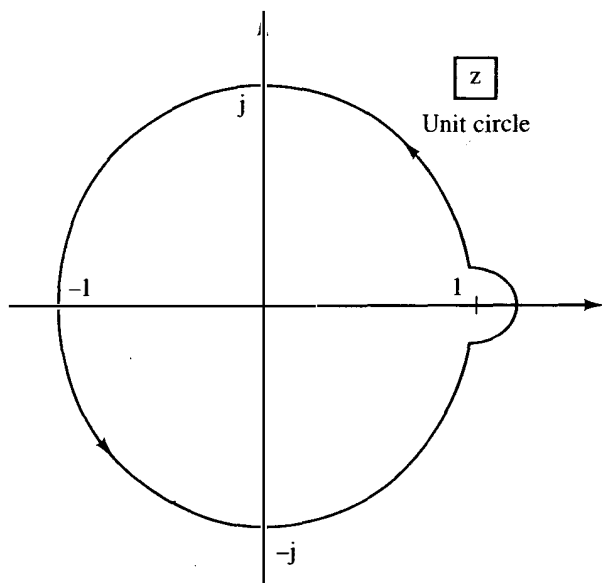


Figure 7-15 Nyquist path in the z -plane.

where N is the number of the clockwise encirclements of the -1 point made by the Nyquist diagram of $\overline{GH}(z)$. The minus sign appears in (7-24) since the Nyquist path is counterclockwise. Now, in general, the order of the numerator and the order of the denominator of $1 + \overline{GH}(z)$ are the same. Let this order be n . Then

$$Z_o + Z_i = n \quad (7-25)$$

$$P_o + P_i = n \quad (7-26)$$

Solving (7-25) and (7-26) for Z_i and P_i , respectively, and substituting the results in (7-24), we obtain

$$N = Z_o - P_o \quad (7-27)$$

Thus the Nyquist criterion is given by (7-27), with the Nyquist path shown in Figure 7-15. The Nyquist diagram for the system of Figure 7-11b is obtained by plotting $\overline{GH}(z)$ for values of z on the unit circle. The subscripts in (7-27) are usually omitted; then

$$N = Z - P \Rightarrow Z = N + P \quad (7-28)$$

where N = number of clockwise encirclements of the -1 point

Z = number of zeros of the characteristic equation outside the unit circle

P = number of poles of the open-loop function, or equivalently poles of the characteristic equation, that are outside the unit circle

Example 7.10

The Nyquist criterion will now be illustrated using the system of Figure 7-16. From Example 6.4,

$$G(z) = \frac{0.368z + 0.264}{(z - 1)(z - 0.368)}$$

The Nyquist path and the Nyquist diagram are both shown in Figure 7-17. The detour around the $z = 1$ point on the Nyquist path (Figure 7-17a) is necessary since $G(z)$ has a pole at this point. On this detour,

$$z = 1 + \rho e^{j\theta}, \quad \rho \ll 1$$

and

$$G(z)|_{z=1+\rho e^{j\theta}} \approx \frac{0.632}{\rho e^{j\theta}(0.632)} = \frac{1}{\rho} \angle -\theta$$

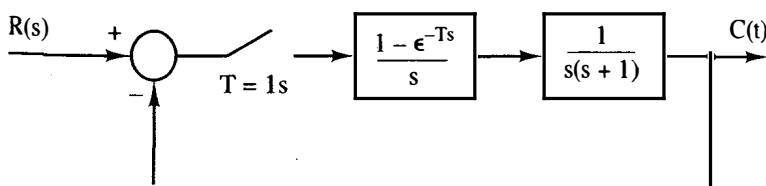


Figure 7-16 System for Example 7.10.

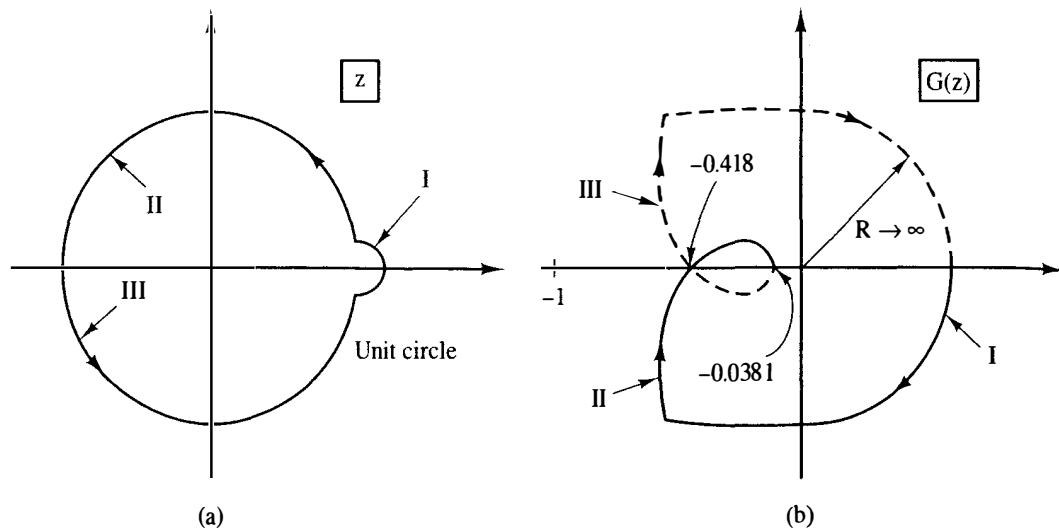


Figure 7-17 Nyquist path and Nyquist diagram for Example 7.10.

Thus this detour generates the large arc on the Nyquist diagram (Figure 7-17b). For z on the unit circle,

$$G(z)|_{z = e^{j\omega T}} = \frac{0.368e^{j\omega T} + 0.264}{(e^{j\omega T} - 1)(e^{j\omega T} - 0.368)}$$

In this equation ω varies from $-\omega_s/2$ to $\omega_s/2$. Since $G(e^{j\omega T})$ for $0 > \omega > -\omega_s/2$ is the complex conjugate of $G(e^{j\omega T})$ for $0 < \omega < \omega_s/2$, it is necessary to calculate $G(e^{j\omega T})$ only for $0 < \omega < \omega_s/2$. This calculation results in the frequency response for $G(z)$. Note that $G(-1) = -0.0381$.

The calculation of the Nyquist diagram for the portion of the diagram labeled II in Figure 7-17b may be accomplished in several ways. We may use

$$G^*(s)|_{s = j\omega} \quad \text{or} \quad G(z)|_{z = e^{j\omega T}}$$

as the basis for calculation. Generally, computer calculations are required to evaluate the frequency response.

We see from Figure 7-17 that $N = 0$, since the Nyquist diagram does not encircle the -1 point. In addition, $P = 0$ since $G(z)$ has no poles outside the Nyquist path. Hence $Z = (N + P) = 0$, and the system is stable.

Suppose that a gain K is added to the plant. The system can be forced into instability by increasing this gain K by a factor of $1/0.418$, or 2.39 . However, the same system without sampling, as shown in Example 7.9, is stable for all positive values of gain K . As stated earlier, the destabilizing effect of the sampling can be seen from the phase lag introduced by the sampler and data hold.

As an additional point in Example 7.10, the analog system *model* is stable for all positive values of plant gain K . This does not carry over to the physical system that is being modeled. The second-order model will be accurate only over limited signal levels. A large increase in gain within the loop will result in large signals, and in general, the linear second-order model will no longer be accurate.

TABLE 7-4 FUNCTIONS FOR CALCULATING THE NYQUIST DIAGRAM

Open-loop function	Range of variable
$\overline{GH}^*(s)$	$s = j\omega, \quad 0 \leq \omega \leq \omega_s/2$
$\overline{GH}(z)$	$z = e^{j\omega T}, \quad 0 \leq \omega T \leq \pi$
$\overline{GH}(w)$	$w = j\omega_w, \quad 0 \leq \omega_w < \infty$

Example 7.11

Consider again the system of Example 7.10. For this system

$$G(z) = \frac{0.368z + 0.264}{z^2 - 1.368z + 0.368}$$

Then $G(w)$ is given by, from Example 7.3,

$$G(w) = \frac{0.368 \left[\frac{1 + 0.5w}{1 - 0.5w} \right] + 0.264}{\left[\frac{1 + 0.5w}{1 - 0.5w} \right]^2 - 1.368 \left[\frac{1 + 0.5w}{1 - 0.5w} \right] + 0.368} = \frac{-0.0381(w - 2)(w + 12.14)}{w(w + 0.924)}$$

The Nyquist diagram can be obtained from this equation by allowing w to assume values from $j0$ to $j\infty$. Since $G(w)$ has a pole at the origin, the Nyquist path must detour around this point. The Nyquist diagram generated by $G(w)$ is identical to that generated using $G(z)$, as shown in Figure 7-17.

Three different transfer functions may be used to generate the Nyquist diagram: $\overline{GH}^*(s)$, $\overline{GH}(z)$, and $\overline{GH}(w)$. Of course, the Nyquist diagrams generated will be identical in each case. Table 7-4 gives the functions and the range of variables required for the calculation of the Nyquist diagram. The range of variables gives only the upper half of the Nyquist path. If a pole of the open-loop function occurs in the range of the variable, a detour must be made around the point that the pole occurs.

For a stable continuous-time system, the *gain margin* is defined as the factor by which the gain must change to force the system to marginal stability. The *phase margin* is defined as the angle through which the Nyquist diagram must be rotated such that the diagram intersects the -1 point. The gain and phase margin definitions for stable discrete-time systems are exactly the same; that is, in the Nyquist diagram shown in Figure 7-18, the gain margin is $1/a$, and the phase margin is ϕ_m .

At this point a discussion of certain properties of pulse transfer functions is in order. For the system of Figure 7-19a,

$$C(z) = G(z)E(z)$$

where we can express $G(z)$ as

$$G(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_0} \quad (7-29)$$

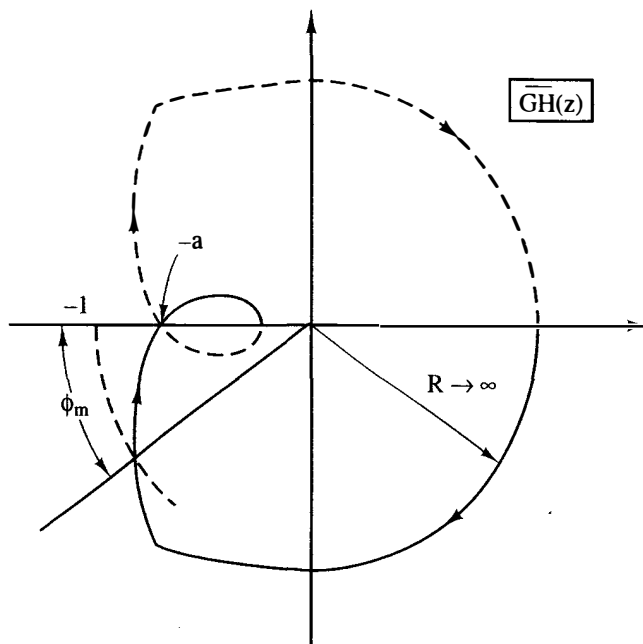


Figure 7-18 Nyquist diagram illustrating stability margins.

By dividing the denominator of $G(z)$ into its numerator, we can express $G(z)$ as

$$G(z) = g(n-m)z^{-(n-m)} + g(n-m+1)z^{-(n-m+1)} + \dots \quad (7-30)$$

The *discrete unit impulse function* is defined by the transform

$$E(z) = 1 \quad (7-31)$$

Thus the discrete unit impulse function is a number sequence $\{e(k)\}$ that has a value of unity for $k = 0$, and a value of zero for $k = 1, 2, 3, \dots$. Hence, if in Figure 7-19, $E(z)$ is the unit impulse function, then

$$C(z) = G(z)E(z) = G(z) \quad (7-32)$$

Thus (7-30) is the system impulse response.

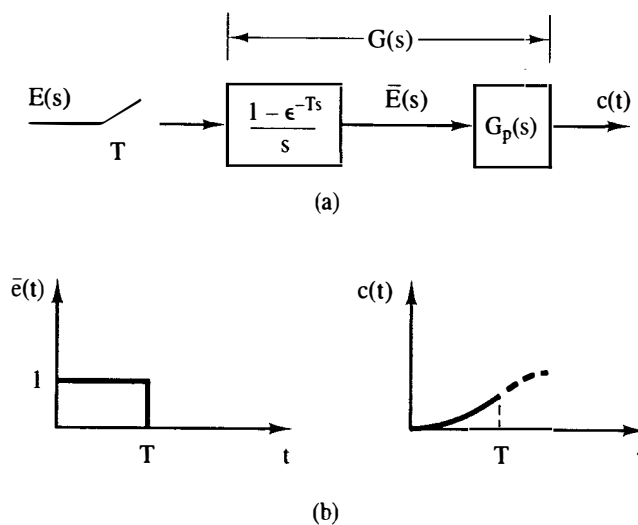


Figure 7-19 Discrete unit pulse response.

Figure 7-19b shows $\bar{e}(t)$ for the case that $E(z)$ is the unit impulse function. For most physical systems we would expect a response $c(t)$ as shown in the figure. The plant does not respond instantaneously to the input, but it will have responded to some nonzero value at $t = T$. Hence we may express $C(z)$ as

$$C(z) = c(1)z^{-1} + c(2)z^{-2} + \dots \quad (7-33)$$

Comparing (7-33) with (7-30), we see that, in general,

$$n - m = 1 \quad (7-34)$$

Then, for a physical system, we generally expect the order of the numerator of its pulse transfer function to be one less than the order of the denominator. If the plant can respond instantaneously, the order of the numerator will be equal to the order of the denominator. If the order of the numerator is greater than the order of the denominator, the plant will respond prior to the application of the input. This case, of course, is not physically realizable; the plant is said to be *noncausal*.

7.8 THE BODE DIAGRAM

The convenience of frequency-response plots for continuous-time systems in the form of the Bode diagram in both analysis and design stems from the straight-line approximations that are made, and these straight-line approximations are based on the independent variable, $j\omega$, being imaginary [1,2]. Thus Bode diagrams of discrete-time systems may be plotted, using straight-line approximations, provided that the w -plane form of the transfer function is used. For convenience, a summary of the first-order terms used in the construction of a Bode diagram is given in Table 7-5 and Figure 7-20. Since terms with complex zeros or poles are not amenable to straight-line approximations, these terms have been omitted.

TABLE 7-5 SUMMARY OF TERMS EMPLOYED IN A BODE DIAGRAM

1. A *constant term* K . When this term is present, the log magnitude plot is shifted up or down by the amount $20 \log_{10} K$.
2. The term $j\omega_w$ or $1/j\omega_w$. If the term $j\omega_w$ is present, the log magnitude is $20 \log_{10} \omega_w$, which is a straight line with a slope of $+20$ dB/decade, and the phase is constant at $+90^\circ$. If the term $1/j\omega_w$ is present, the log magnitude is $-20 \log_{10} \omega_w$, which is a straight line with a slope of -20 dB/decade, and the phase is constant at -90° . The Bode plots for these terms are shown in Figure 7-20.
3. The term $(1 + j\omega_w \tau)$ or $[1/(1 + j\omega_w \tau)]$. The term $(1 + j\omega_w \tau)$ has a log magnitude of $20 \log_{10} \sqrt{1 + \omega_w^2 \tau^2}$ which can be approximated as $20 \log_{10} 1 = 0$ when $\omega_w \tau \ll 1$ and as $20 \log_{10} \omega_w \tau$, $\omega_w \tau \gg 1$. The corner or "break" frequency is $\omega_w = 1/\tau$. The phase is given by the expression $\tan^{-1} \omega_w \tau$. The term $[1/(1 + j\omega_w \tau)]$ is handled in a similar manner. The Bode plots for these functions are shown in Figure 7-20.

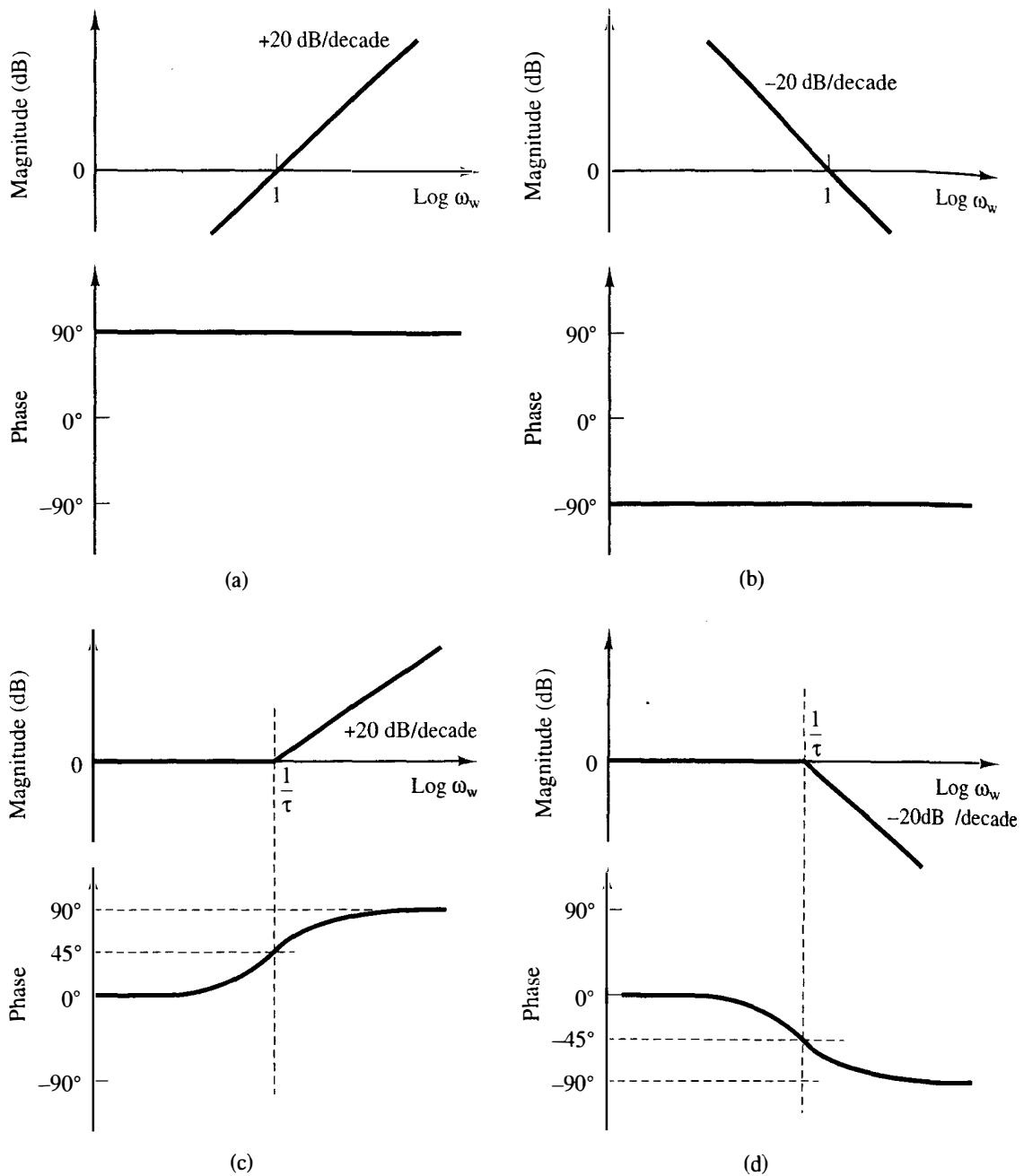


Figure 7-20 Short summary of terms employed in Bode diagrams: (a) Bode plot for $j\omega_w$; (b) Bode plot for $1/j\omega_w$; (c) Bode plot for $1 + j\omega_w\tau$; (d) Bode plot for $1/(1 + j\omega_w\tau)$.

Let us now employ the Bode diagram in the analysis of a familiar example.

Example 7.12



Consider again the system of Example 7.3. For this system,

$$G(w) = -\frac{0.0381(w - 2)(w + 12.14)}{w(w + 0.924)}$$

and

$$\begin{aligned}
 G(j\omega_w) &= -\frac{0.0381(j\omega_w - 2)(j\omega_w + 12.14)}{j\omega_w(j\omega_w + 0.924)} \\
 &= \frac{-\left(j\frac{\omega_w}{2} - 1\right)\left(\frac{j\omega_w}{12.14} + 1\right)}{j\omega_w\left(\frac{j\omega_w}{0.924} + 1\right)}
 \end{aligned}$$

Note that the numerator break frequencies are $\omega_w = 2$ and $\omega_w = 12.14$ and the denominator break frequencies are $\omega_w = 0$ and $\omega_w = 0.924$. The Bode diagram for this system, using straight-line approximations, is shown in Figure 7-21. Both the gain and phase margins of the system are shown on the diagram.

As stated in Example 7-10, this system can be made unstable by increasing the gain. This can also be seen in Figure 7-21. Increasing the gain is equivalent to shifting the entire magnitude curve vertically upward. Therefore, if the gain is increased by an amount equal to the gain margin, a condition of marginal stability will exist. A

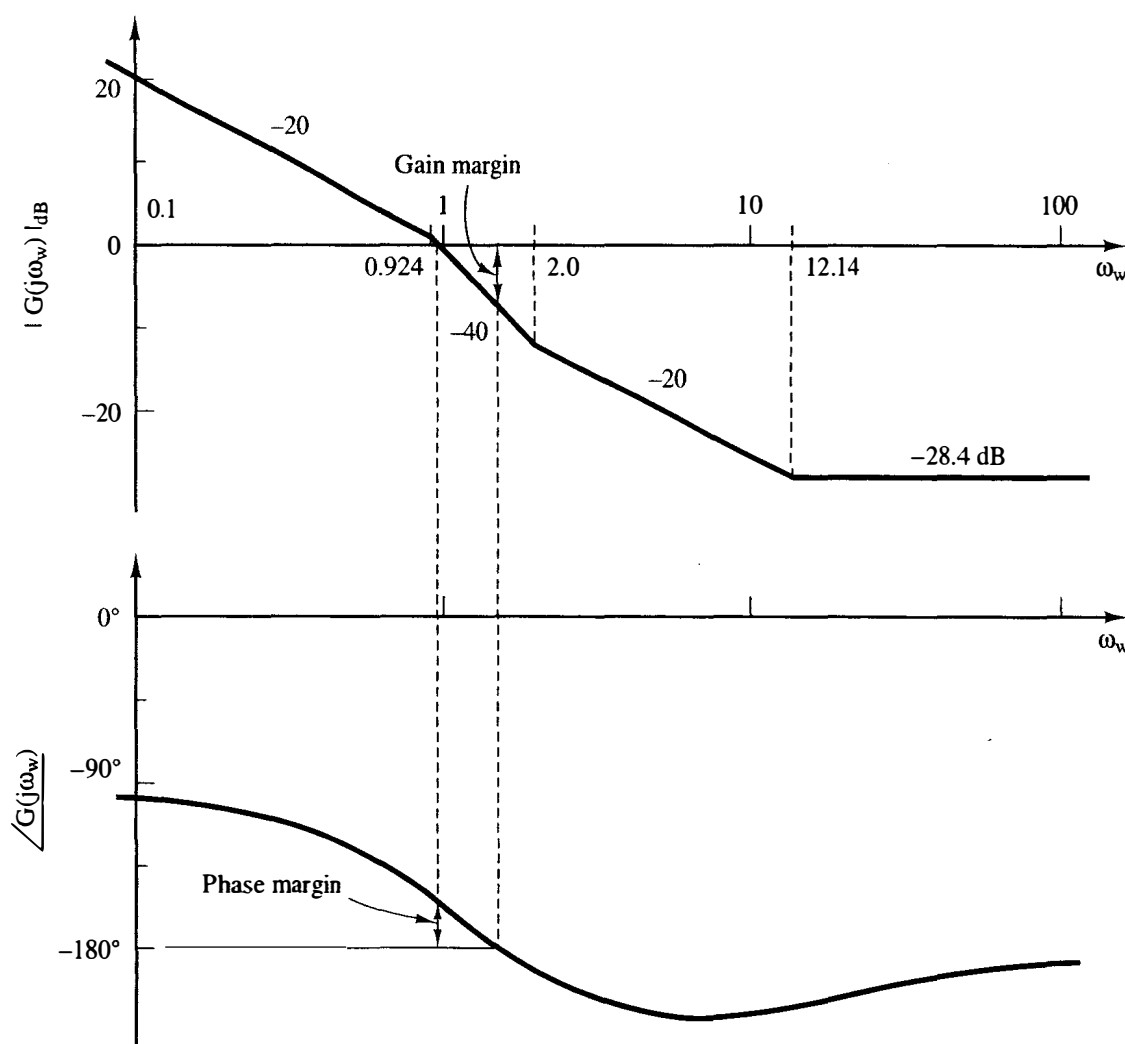


Figure 7-21 Bode diagram for Example 7.12.

MATLAB program that solves for and plots a Bode diagram for this example is given by

```
ww = logspace(-1,1,10);
n = [-0.0381 -0.386 0.924];
d = [1 0.924 0];
[mag,phase,ww] = bode(n,d,ww);
db = 20*log10(mag);
disp(' omegaw mag dB phase')
[ww',mag, db,phase]
pause
% for plot:
subplot(211), semilogx(ww,db)
title('Bode diagram'); xlabel('w-plane frequency');
ylabel('dB'); grid
subplot(212), semilogx(ww,phase)
xlabel('w-plane frequency'); ylabel('phase'); grid
pause, subplot (111)
```

In this program, ww is w -plane frequency, mag is the magnitude of $G(j\omega_w)$, db is its magnitude in dB, and $phase$ is its phase.

While this section presents an approximate technique for generating Bode diagrams, in practice these diagrams are usually plotted from computer-calculated data. The frequency response may be calculated from either $\overline{GH}^*(s)$ or $\overline{GH}(z)$, and for each s -plane frequency ω , the w -plane frequency ω_w is obtained from (7-10); that is,

$$\omega_w = \frac{2}{T} \tan\left(\frac{\omega T}{2}\right)$$

The frequency response as calculated from $\overline{GH}^*(s)$ or $\overline{GH}(z)$ is then plotted versus ω_w . Thus it is unnecessary to construct the function $G(w)$.

The gain-phase plot of the frequency response of a system is a plot of the same information shown by a Bode diagram, plotted on different axes. For the gain-phase plot, the frequency response is plotted as gain versus phase on rectangular axes, with frequency as a parameter. Thus the frequency response of a discrete-time system may be plotted on gain-phase axes, as well as on a Bode diagram. As an example, the gain-phase plot of the system of Example 7.12 is shown in Figure 7-22.

7.9 INTERPRETATION OF THE FREQUENCY RESPONSE

Throughout this chapter the term *frequency response* has been used in relation to discrete-time systems. The physical meaning of a system frequency response in relation to continuous-time systems is well known. In this section the physical meaning of the frequency response of a discrete-time system is developed.

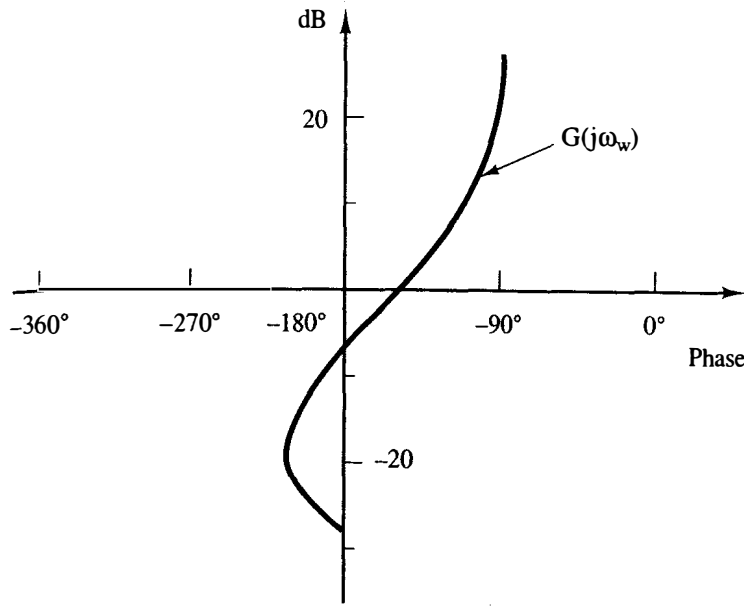


Figure 7-22 Gain-phase plot for Example 7.12.

Consider a discrete-time system described by

$$C(z) = G(z)E(z)$$

and suppose that the system input is a sampled sine wave; that is,

$$E(z) = \mathcal{Z}[\sin \omega t] = \frac{z \sin \omega T}{(z - \epsilon^{j\omega T})(z - \epsilon^{-j\omega T})}$$

Then

$$C(z) = \frac{G(z)z \sin \omega T}{(z - \epsilon^{j\omega T})(z - \epsilon^{-j\omega T})} = \frac{k_1 z}{z - \epsilon^{j\omega T}} + \frac{k_2 z}{z - \epsilon^{-j\omega T}} + C_g(z) \quad (7-35)$$

where $C_g(z)$ are those components of $C(z)$ that originate in the poles of $G(z)$. If the system is stable, these components of $c(nT)$ will tend to zero with increasing time, and the system steady-state response is given by

$$C_{ss}(z) = \frac{k_1 z}{z - \epsilon^{j\omega T}} + \frac{k_2 z}{z - \epsilon^{-j\omega T}}$$

Now, from (7-35),

$$k_1 = \frac{G(\epsilon^{j\omega T}) \sin \omega T}{\epsilon^{j\omega T} - \epsilon^{-j\omega T}} = \frac{G(\epsilon^{j\omega T})}{2j} \quad (7-36)$$

Expressing $G(\epsilon^{j\omega T})$ as

$$G(\epsilon^{j\omega T}) = |G(\epsilon^{j\omega T})| \epsilon^{j\theta}$$

we see that (7-36) becomes

$$k_1 = \frac{|G(\epsilon^{j\omega T})| \epsilon^{j\theta}}{2j} \quad (7-37)$$

Since k_2 is the complex conjugate of k_1 , then

$$k_2 = \frac{|G(\epsilon^{j\omega T})|\epsilon^{-j\theta}}{2(-j)} = \frac{-|G(\epsilon^{j\omega T})|\epsilon^{-j\theta}}{2j} \quad (7-38)$$

Thus, from (7-36), (7-37), and (7-38),

$$\begin{aligned} c_{ss}(kT) &= k_1(\epsilon^{j\omega T})^k + k_2(\epsilon^{-j\omega T})^k \\ &= |G(\epsilon^{j\omega T})| \frac{\epsilon^{j(\omega kT + \theta)} - \epsilon^{-j(\omega kT + \theta)}}{2j} \\ &= |G(\epsilon^{j\omega T})| \sin(\omega kT + \theta) \end{aligned} \quad (7-39)$$

From the development above it is seen that if the input to a stable discrete-time system is a sinusoid of frequency ω , the steady-state system response is also sinusoidal at the same frequency. The amplitude of the response is equal to the amplitude of the input multiplied by $|G(\epsilon^{j\omega T})|$, and the phase of the response is equal to the phase of the input plus the angle of $G(\epsilon^{j\omega T})$. Thus it is seen that $G(\epsilon^{j\omega T})$ is the true system frequency response at the sampling instants.

7.10 CLOSED-LOOP FREQUENCY RESPONSE

In this chapter many of the techniques presented for closed-loop stability analysis are based on the system open-loop frequency response. Of course, the system transfer characteristics are determined by the closed-loop frequency response. In this section the closed-loop frequency response of a control system is related graphically to its open-loop frequency response. The resultant graph is important conceptually, especially in the design of control systems; this graph is useful in illustrating the meaning of gain and phase margins as relative-stability parameters. However, if we want to calculate the closed-loop frequency response, we do not use graphical techniques; instead, we use computer software.

Consider the discrete-time system shown in Figure 7-23a. For this system,

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1 + G(z)}$$

The closed-loop frequency response is given by

$$\frac{C(\epsilon^{j\omega T})}{R(\epsilon^{j\omega T})} = \frac{G(\epsilon^{j\omega T})}{1 + G(\epsilon^{j\omega T})} \quad (7-40)$$

Suppose that $G(\epsilon^{j\omega T})$ is as shown in the polar plot in Figure 7-23b. Then the numerator and the denominator of (7-40) are the vectors shown for the frequency $\omega = \omega_1$, and the frequency response at ω_1 is the ratio of these vectors. Let the ratio of these vectors be denoted as

$$\frac{C(\epsilon^{j\omega_1 T})}{R(\epsilon^{j\omega_1 T})} = \frac{|G(\epsilon^{j\omega_1 T})|\epsilon^{j\theta}}{|1 + G(\epsilon^{j\omega_1 T})|\epsilon^{j\beta}} = M\epsilon^{j(\theta - \beta)} = M\epsilon^{j\phi} \quad (7-41)$$

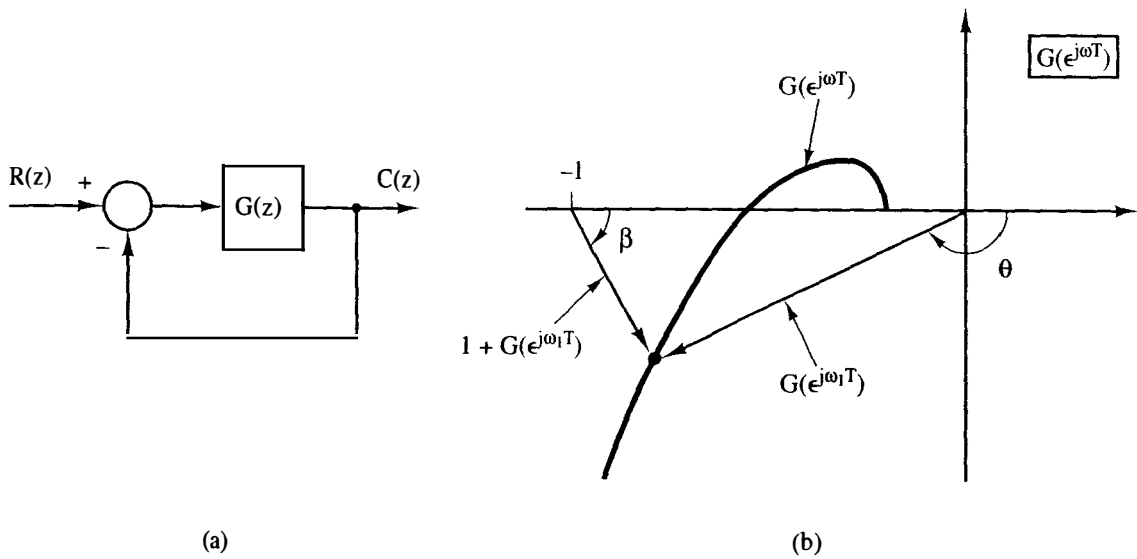


Figure 7-23 Determining closed-loop frequency response from open-loop frequency response.

The locus of points in the $G(e^{j\omega T})$ plane for which the magnitude of the closed-loop frequency response, M , is a constant is called a constant magnitude locus, or a constant M circle. To see that these loci are in fact circles, consider the following development. Let

$$G(e^{j\omega T}) = X + jY \quad (7-42)$$

Then, from (7-40) and (7-41),

$$M^2 = \frac{X^2 + Y^2}{(1 + X)^2 + Y^2}$$

Hence

$$X^2(1 - M^2) - 2M^2X - M^2 + (1 - M^2)Y^2 = 0 \quad (7-43)$$

For $M \neq 1$, we can express this relationship as

$$\left[X + \frac{M^2}{M^2 - 1} \right]^2 + Y^2 = \frac{M^2}{(M^2 - 1)^2} \quad (7-44)$$

This relationship is the equation of a circle of radius $|M/(M^2 - 1)|$ with center at $X = -M^2/(M^2 - 1)$ and $Y = 0$. For $M = 1$, (7-43) yields $X = -\frac{1}{2}$, which is a straight line. Figure 7-24 illustrates the constant M circles.

The loci of points of constant phase are also circles. It is seen from (7-41) and (7-42) that

$$\phi = \theta - \beta = \tan^{-1}\left(\frac{Y}{X}\right) - \tan^{-1}\left(\frac{Y}{1 + X}\right)$$

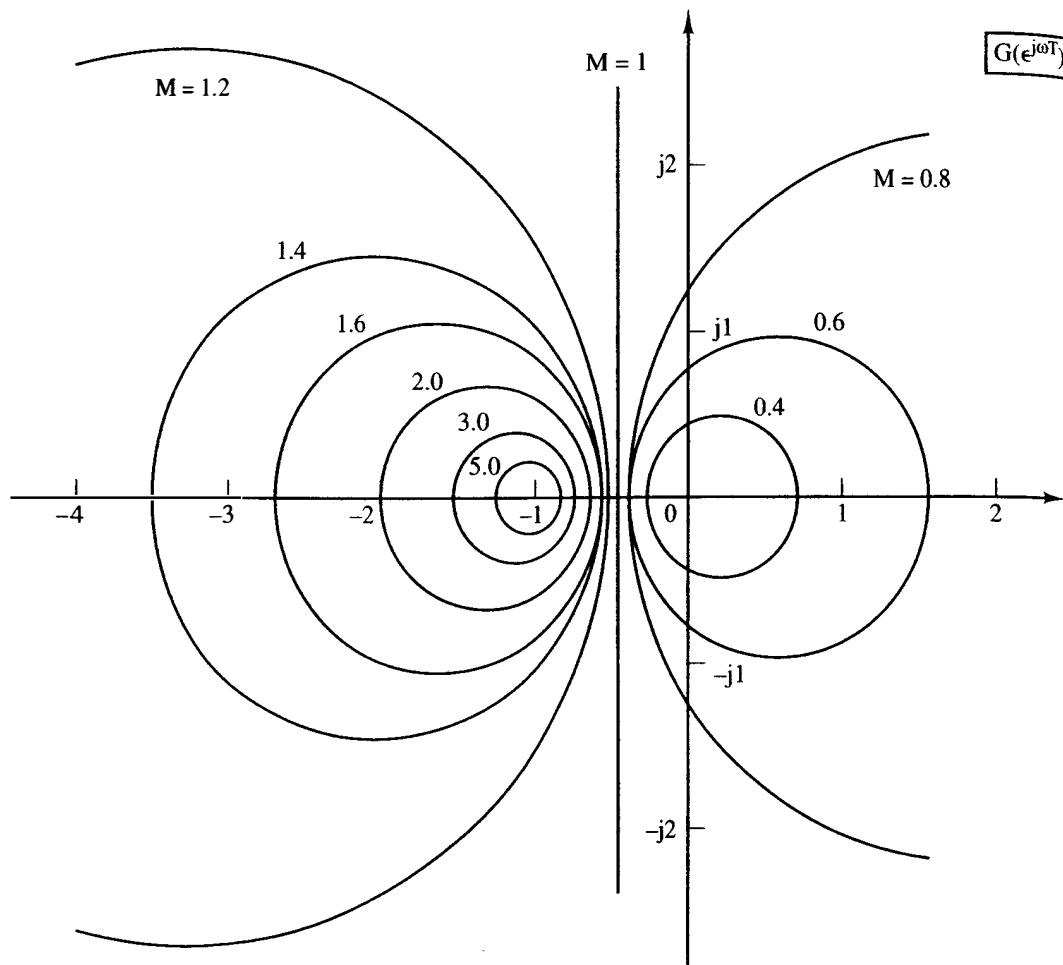


Figure 7-24 Constant magnitude circles.

Then, letting $N = \tan \phi$,

$$N = \tan(\theta - \beta) = \frac{\tan \theta - \tan \beta}{1 + \tan \theta \tan \beta} = \frac{\frac{Y}{X} - \frac{Y}{1+X}}{1 + \frac{Y}{X} \left[\frac{Y}{1+X} \right]}$$

Hence

$$X^2 + X + Y^2 - \frac{1}{N}Y = 0$$

This equation can then be expressed as

$$\left(X + \frac{1}{2}\right)^2 + \left(Y - \frac{1}{2N}\right)^2 = \frac{1}{4} + \left(\frac{1}{2N}\right)^2 \quad (7-45)$$

This is the equation of a circle with radius of $\sqrt{1/4 + (1/2N)^2}$ with center at $X = -1/2$ and $Y = 1/(2N)$. Figure 7-25 illustrates the constant N (phase) circles. Note that

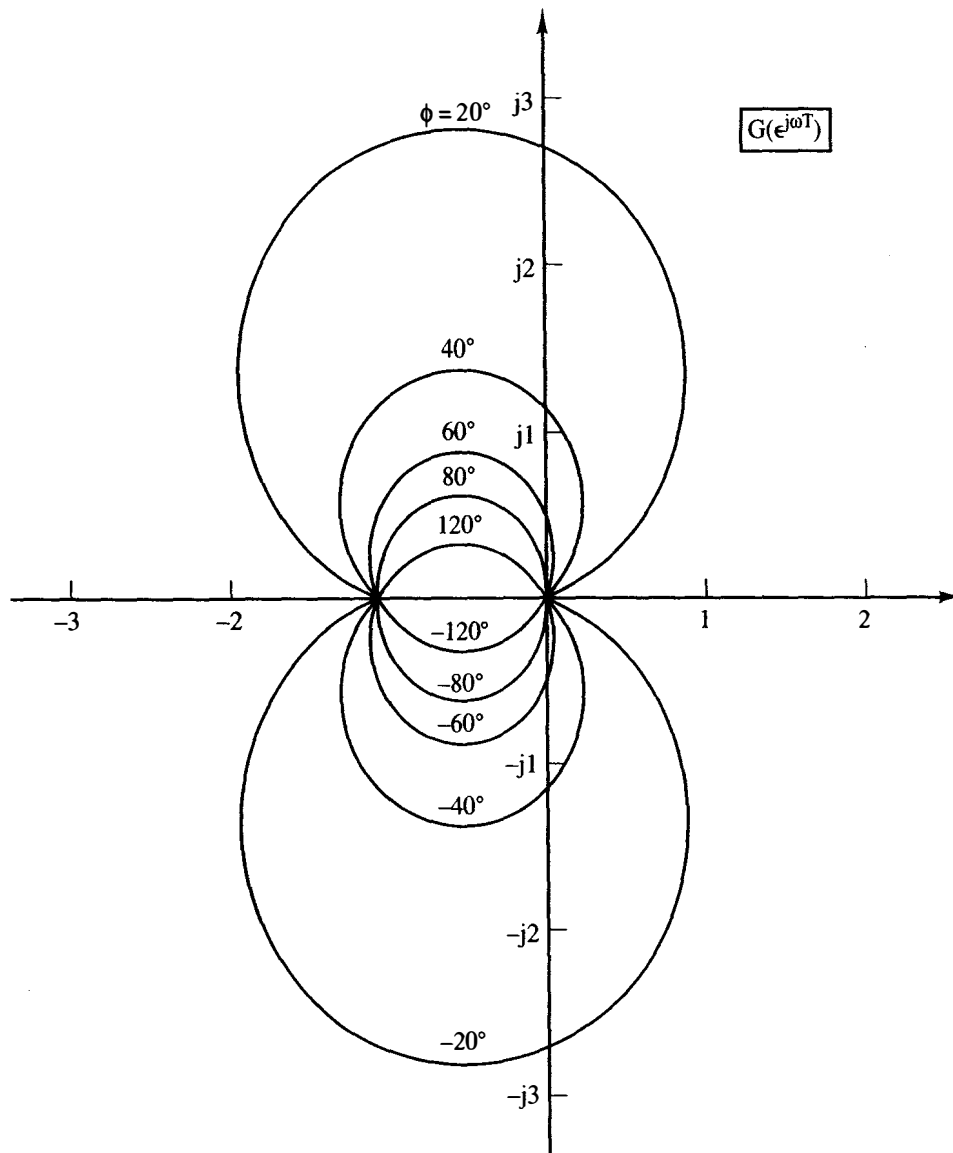


Figure 7-25 Constant phase circles.

accuracy is very limited in the region of the -1 point for Figures 7-24 and 7-25. For this reason the Nichols chart is often used, instead of the constant M and N circles. The Nichols chart will now be discussed.

The gain-phase plane was introduced in Section 7.8 and was shown to be useful in graphically presenting a system frequency response. For example, the frequency response $G(e^{j\omega T})$ presented as a polar plot in Figure 7-23 could also be plotted in the gain-phase plane, as shown in Figure 7-22. Also, the constant M circles of Figure 7-24 and the constant N circles of Figure 7-25 can be plotted in the gain-phase plane. Such a plot is called the Nichols chart [6], and is often used in relating the open-loop frequency response and the closed-loop frequency response. Figure 7-26 illustrates the Nichols chart.

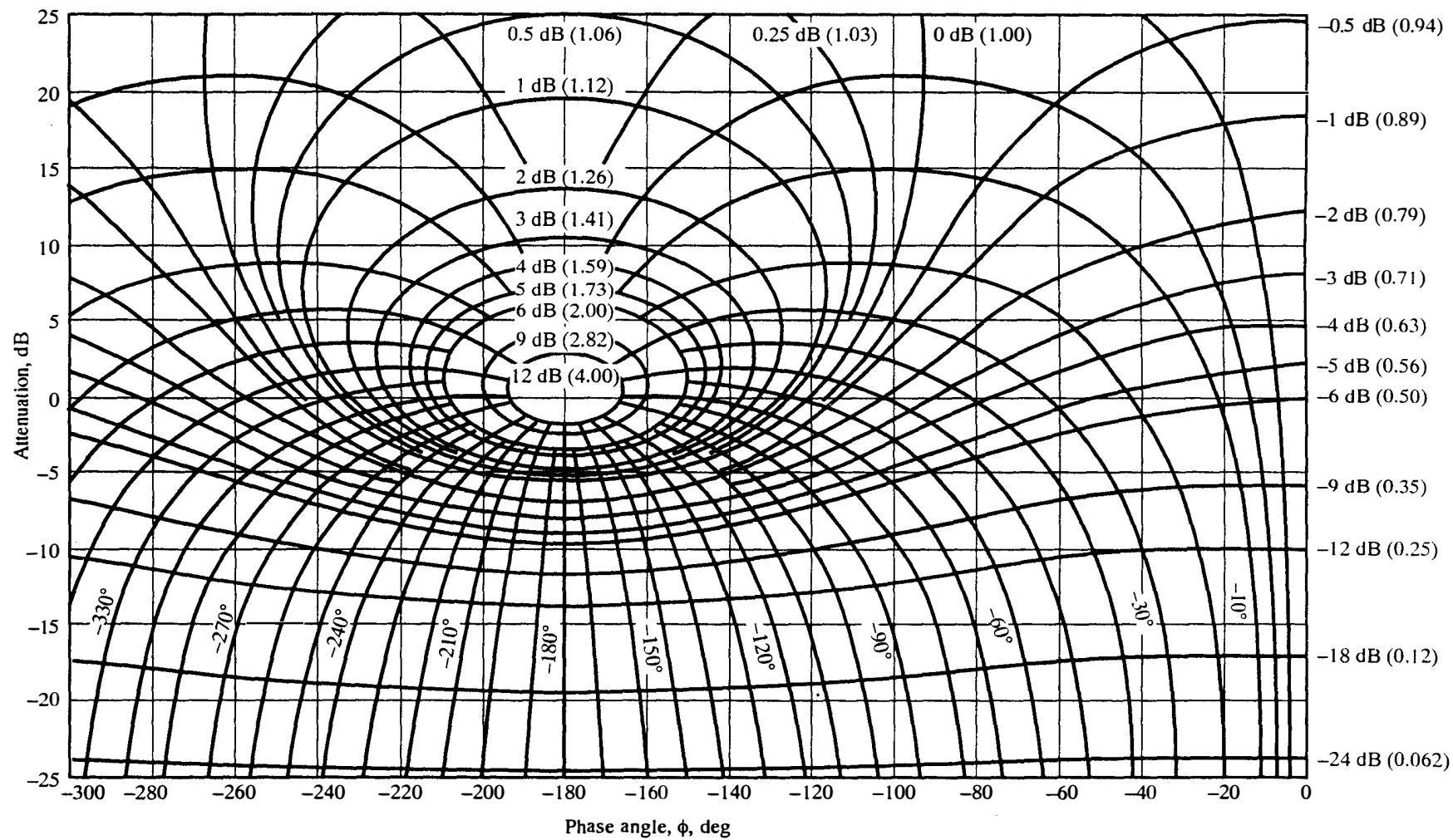


Figure 7-26 Nichols chart.

The frequency response $G(e^{j\omega T})$ is usually computed via computer programs. Thus it is logical to add statements to these programs that will, at the same time, calculate the closed-loop frequency response, rather than using a graphical procedure as described above. The computer procedure is more accurate, saves time, and is not limited to the system of Figure 7-23a. However, as stated above, the constant M and N circles and the Nichols chart are very useful in understanding the effects on the closed-loop frequency response of varying the open-loop frequency response. An example will now be given that illustrates closed-loop frequency response.

Example 7.13

Again the system of Example 6.4 will be considered. For this example, $T = 1$ s and

$$G^*(s) = \left[\frac{1 - e^{-Ts}}{s^2(s + 1)} \right]^*$$

The closed-loop frequency response was calculated by computer and is plotted in Figure 7-27 versus real frequency ω . To illustrate the effects on the step response of the closed-loop frequency response, the closed-loop frequency response was then calculated for $T = 0.1$ s and is also shown in Figure 7-27. Note that a resonant, or peaking, effect is more pronounced at $T = 1$ s. A more pronounced resonance in the closed-loop frequency response generally indicates more overshoot in the step response. The step responses for the system were obtained by simulation, and are shown in Figure 7-28. The peak overshoot for $T = 1$ s is 45 percent, and that for $T = 0.1$ s is 18 percent.

A resonance in a closed-loop frequency response can also be correlated with the system stability margins. For the system of Figure 7-23a, the denominator of the closed-loop transfer function is the distance from the -1 point to the Nyquist diagram, as shown in Figure 7-23b. If the stability margins are small, the Nyquist diagram passes close to the -1 point, and the denominator of the closed-loop transfer function has a distinct minimum. For this case, the closed-loop system will have a resonance. For the systems of Example 7.13, with $T = 0.1$ s, the phase margin is 50° and the gain margin is 26 dB. For $T = 1$ s, the phase margin is 30° and the gain margin is 7.6 dB. The correlation between the closed-loop frequency response and the time response of a system is investigated in greater detail in Chapter 8.

In summary, we see from both the constant M circles of Figure 7-24 and the Nichols chart of Figure 7-26 that a peaking in the closed-loop frequency response

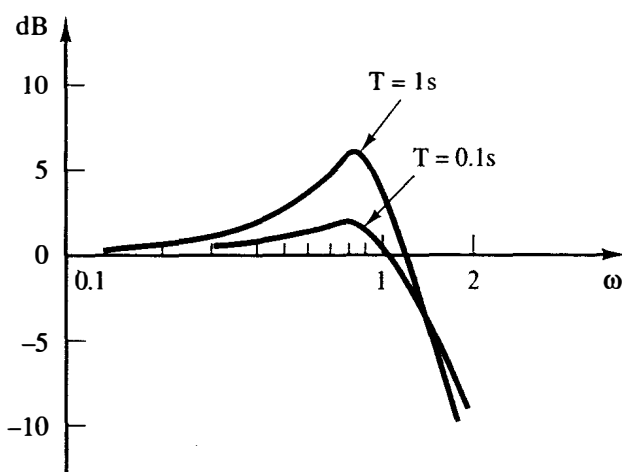


Figure 7-27 Closed-loop frequency response magnitudes for Example 7.13.

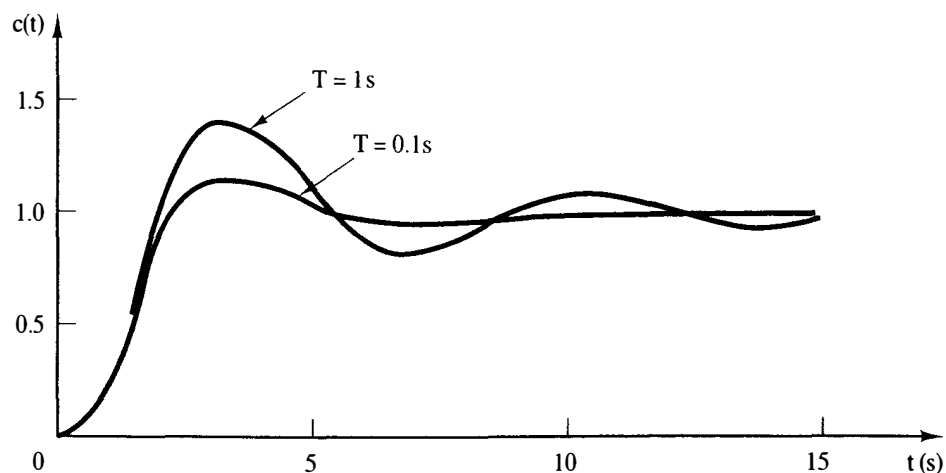


Figure 7-28 Step responses for Example 7.13.

can occur only if the open-loop frequency response passes near the -1 point. Hence any control system that has a significant resonance also has small stability margins. Conversely, any system that has small stability margins will exhibit a significant resonance in its time response.

7.11 SUMMARY

In this chapter a number of techniques for analyzing the stability of discrete-time systems have been presented. It has been shown that many of the methods used in the analysis of continuous-time systems are applicable to sampled-data systems also. The chapter contains a number of examples, and the use of the same system in many of the examples throughout the chapter provides a common thread and basis for comparison among the various stability analysis techniques. Many of the analysis techniques presented in this chapter will be extended to design in Chapter 8.

REFERENCES AND FURTHER READING

1. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2d ed. Englewood Cliffs, NJ: Prentice Hall, 1991.
2. E. I. Jury, *Theory and Application of the z-Transform Method*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.
3. G. F. Franklin and J. D. Powell, *Feedback Control of Dynamic Systems*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1991.
4. R. C. Dorf, *Modern Control Systems*, 5th ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1991.
5. C. R. Wylie, Jr., *Advanced Engineering Mathematics*, 4th ed. New York: McGraw-Hill Book Company, 1975.
6. H. M. James, N. B. Nichols, and R. S. Phillips, *Theory of Servomechanisms*. New York: McGraw-Hill Book Company, 1947.

7. J. A. Cadzow and H. R. Martens, *Discrete-Time and Computer Control Systems*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1970.
8. W. R. Evans, *Control System Dynamics*. New York: McGraw-Hill Book Company, 1954.
9. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1988.

PROBLEMS

- 7-1. Assume that for the system of Figure 7-1, the system closed-loop transfer-function pole p_1 is repeated such that the system characteristic equation is given by

$$(z - p_1)^r(z - p_{r+1})(z - p_{r+2}) \cdots (z - p_n) = 0$$

where r is an integer. Show that the requirement for system stability is that the magnitudes of all poles of the closed-loop transfer function are less than unity, that is, $|p_i| < 1, i = 1, r + 1, r + 2, \dots, n$.

- 7-2. The system of Example 7.1 and Figure 7-3 has two samplers. The system characteristic equation is derived in Example 7.1 as

$$1 + G_1(z)G_2(z) + \overline{G_2H}(z) = 0$$

Show that the same characteristic equation is obtained by opening the system at the second sampler.

- 7-3. (a) The unit-step response of a discrete system is the system response $c(k)$ with the input $r(k) = 1$ for $k \geq 0$. Show that if the discrete system is stable, the unit-step response, $c(k)$, approaches a constant as $k \rightarrow \infty$. [Let $T(z)$ be the closed-loop system transfer function. Assume that the poles of $T(z)$ are distinct (no repeated poles).]
 (b) Find the conditions on the closed-loop system transfer function $T(z)$ such that the unit-step response approaches zero as $k \rightarrow \infty$.
 (c) The discrete unit-impulse response of a discrete system is the system response $c(k)$ with the input $r(k) = 1$ for $k = 0$ and $r(k) = 0$ for $k \geq 1$. Show that if the discrete system is stable, the unit impulse response, $c(k)$, approaches zero. [Let $T(z)$ be the closed-loop system transfer function. Assume that the poles of $T(z)$ are distinct (no repeated poles).]
 7-4. Consider a sampled-data system with $T = 0.5$ s and the characteristic equation given by

$$(z - 0.9)(z - 0.8)(z^2 - 1.9z + 1.0) = 0$$

- (a) Find the terms in the system natural response.
 - (b) A discrete LTI system is stable, unstable, or marginally stable. Identify the type of stability for this system.
 - (c) The natural response of this system contains an undamped sinusoidal response term of the form $A \cos(\omega kT + \theta)$. Find the frequency ω of this term.
- 7-5. Given below are the characteristic equations of certain discrete systems.
- | | |
|---|---|
| (i) $z^2 - 1.1z + 0.3 = 0$ | (ii) $z^2 - z + 0.25 = 0$ |
| (iii) $z^2 - 0.1z - 0.3 = 0$ | (iv) $z^2 - 0.25 = 0$ |
| (v) $z^2 - 1.6z + 1 = 0$ | (vi) $z^2 - 2.0z + 0.99 = 0$ |
| (vii) $z^3 - 2.2z^2 + 1.55z - 0.35 = 0$ | (viii) $z^3 - 1.9z^2 + 1.4z - 0.45 = 0$ |
- (a) Use the Jury test to determine the stability of each of the systems.
 - (b) List the natural-response terms for each of the systems. A computer may be used to find the roots of the characteristic equations.

- (c) For those systems in part (a) that are found to be either unstable or marginally stable, list the natural-response terms in part (b) that yield these results.

7-6. Consider the system of Figure P7-6 with $T = 1$ s. Let the digital controller be a variable gain K such that $D(z) = K$. Hence $m(kT) = Ke(kT)$.

- Write the closed-loop system characteristic equation.
- Determine the range of K for which the system is stable.
- Suppose that K is set to the lower limit of the range in part (b) such that the system is marginally stable. Find the natural-response term that illustrates the marginal stability.
- Repeat part (c) for the upper limit of the range of K .
- Verify the results of this problem by digital computation.

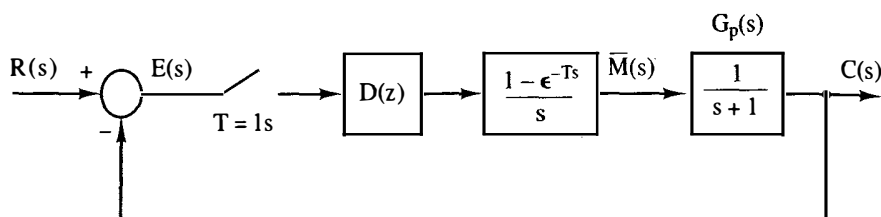


Figure P7-6 System for Problem 7-6.

- 7-7. Consider the system of Figure P7-6 with $T = 1$ s. Let the digital controller be a variable gain K such that $D(z) = K$. Hence $m(kT) = Ke(kT)$.
- Write the closed-loop system characteristic equation.
 - Use the Routh–Hurwitz criterion to determine the range of K for stability.
 - Check the results of part (b) using the Jury test.
 - Determine the location of all roots of the characteristic equation in both the w -plane and the z -plane for the value of $K > 0$ for which the system is marginally stable.
 - Determine the s -plane frequency at which the system will oscillate when marginally stable, for $K > 0$.
 - Consider the system with all sampling removed and with $G_p(s) = K/(s + 1)$. Find the range of K for which the analog system is stable.
 - Comparing the ranges of K from parts (b) and (f), give the effects on stability of adding sampling to the analog system.
 - Verify the results of this problem by digital computation.
- 7-8. Consider the system of Figure P7-6, and let the digital controller be a variable gain K such that $D(z) = K$. Hence $m(kT) = Ke(kT)$.
- Write the closed-loop system characteristic equation as a function of the sample period T .
 - Determine the ranges of $K > 0$ for stability for the sample periods $T = 1$ s, $T = 0.1$ s, and $T = 0.01$ s.
 - Consider the system with all sampling removed and with $G_p(s) = K/(s + 1)$. Find the range of $K > 0$ for which the analog system is stable.
 - Comparing the ranges of K from parts (b) and (c), give the effects on stability of reducing the sample period T .
 - Verify the results of this problem by digital computation.
- 7-9. Consider the temperature control system of Figure P7-9. This system is described in Problem 1-10. For this problem, ignore the disturbance input, let $T = 0.6$ s, and let the

digital controller be a variable gain K such that $D(z) = K$. Hence $m(kT) = Ke(kT)$, where $e(t)$ is the input to the sampler. It was shown in Problem 6-4 that

$$\mathcal{Z}\left[\frac{1 - e^{-Ts}}{s} \frac{2}{s + 0.5}\right] = \frac{1.037}{z - 0.7408}$$

- Write the closed-loop system characteristic equation.
- Use the Routh-Hurwitz criterion to determine the range of K for stability.
- Check the results of part (b) using the Jury test.
- Let $T = 0.06$ s. Find the range of K for which the system is stable.
- Comparing the ranges of K from parts (b) and (d), give the effects on stability of adding sampling to the analog system.
- Verify the results of this problem by digital computation.

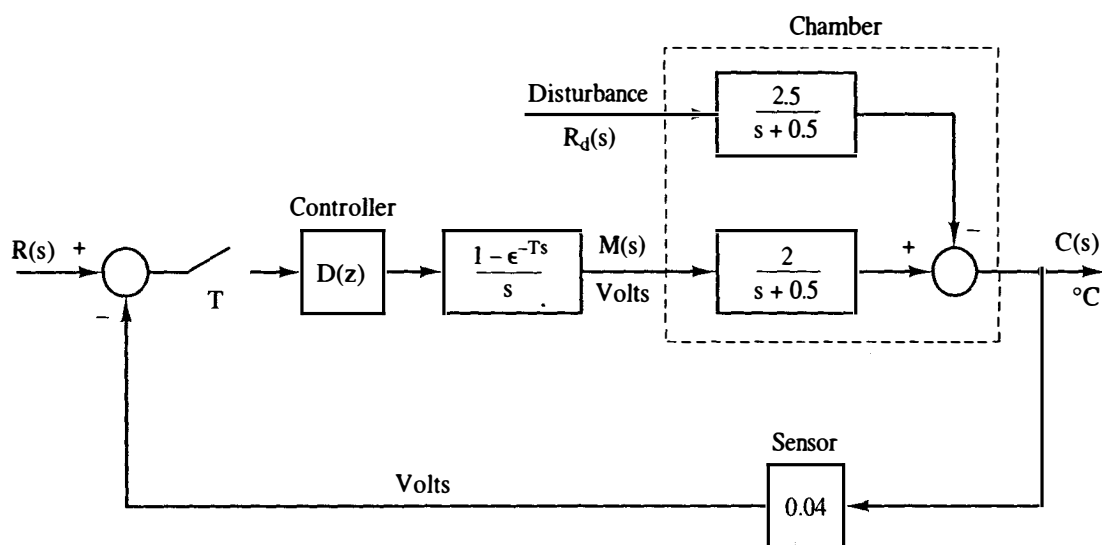


Figure P7-9 Chamber temperature control system.

- 7-10.** Consider the robot arm joint control system of Figure P7-10. This system is described in Problem 1-16. For this problem, $T = 0.1$ s and $D(z) = 1$. It was shown in Problem 6-7 that

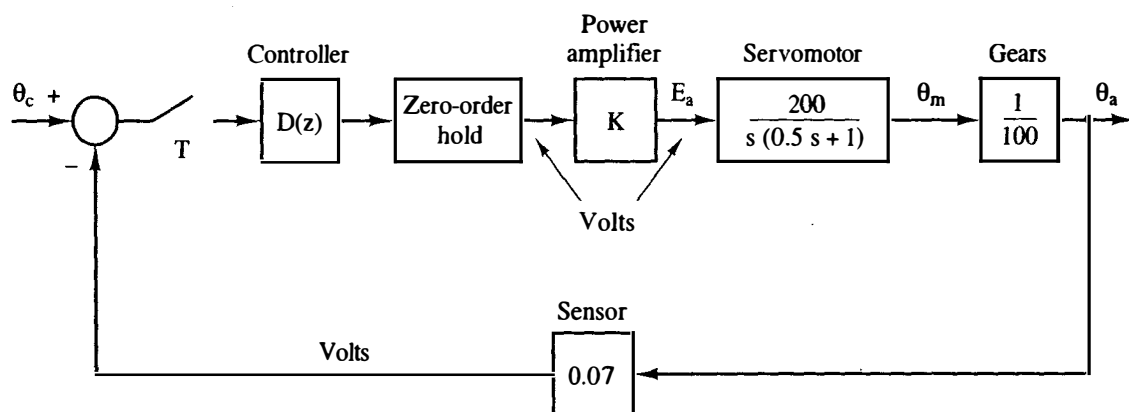


Figure P7-10 Robot arm joint control system.

$$\mathcal{Z} \left[\frac{1 - e^{-Ts}}{s} \frac{4}{s(s+2)} \right] = \frac{0.01873z + 0.01752}{(z-1)(z-0.8187)}$$

- Write the closed-loop system characteristic equation.
 - Use the Routh–Hurwitz criterion to determine the range of K for stability.
 - Check the results of part (b) using the Jury test.
 - Determine the location of all roots of the characteristic equation in both the w -plane and the z -plane for the value of $K > 0$ for which the system is marginally stable.
 - Determine both the s -plane frequency and the w -plane frequency at which the system will oscillate when marginally stable, using the results of part (d).
 - Show that the frequencies in part (e) satisfy (7-10).
 - Verify the results of this problem by digital computation.
- 7-11.** Consider the antenna control system of Figure P7-11. This system is described in Problem 1-7. For this problem, $T = 0.05$ s and $D(z) = 1$. It was shown in Problem 5-15 that

$$\mathcal{Z} \left[\frac{1 - e^{-Ts}}{s} \frac{20}{s(s+6)} \right] = \frac{0.02268z + 0.02052}{(z-1)(z-0.7408)}$$

- Write the closed-loop system characteristic equation.
- Use the Routh–Hurwitz criterion to determine the range of K for stability.
- Check the results of part (b) using the Jury test.
- Determine the location of all roots of the characteristic equation in both the w -plane and the z -plane for the value of $K > 0$ for which the system is marginally stable.
- Determine both the s -plane frequency and the w -plane frequency at which the system will oscillate when marginally stable, using the results of part (d).
- Show that the frequencies in part (e) satisfy (7-10).
- Verify the results of this problem by digital computation.

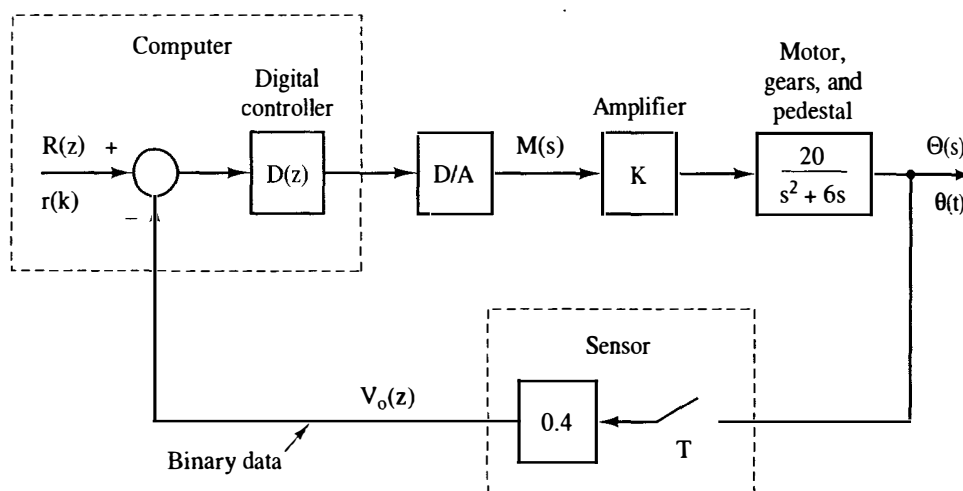


Figure P7-11 Block diagram for an antenna control system.

- 7-12.** Consider the satellite control system of Figure P7-12. This system is described in Problem 1-12. For this problem, $T = 0.1$ s, $J = 0.1$, $H_k = 0.02$, and $D(z) = 1$. From the z -transform table,

$$\mathcal{Z} \left[\frac{1 - e^{-Ts}}{s} \frac{10}{s^2} \right] = \frac{0.05(z+1)}{(z-1)^2}$$

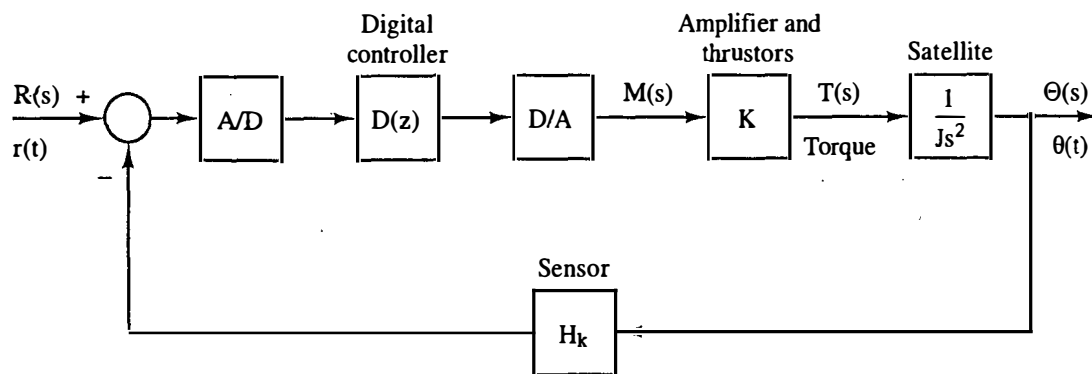


Figure P7-12 Block diagram for a satellite control system.

- Write the closed-loop system characteristic equation.
- Use the Routh–Hurwitz criterion to show that the system is unstable for all K .
- Check the results of part (b) using the Jury test.

7-13. For the system of Figure P7-13, $T = 2$ s and

$$G(z) = \frac{K(z + 0.8)}{(z - 1)(z - 0.6)}$$

- Determine the range of K for stability using the Routh–Hurwitz criterion.
- Determine the range of K for stability using the Jury test.
- Show that the upper limit of K for stability in part (a) yields a marginally stable system.
- Show that the upper limit of K for stability in part (b) yields a marginally stable system.

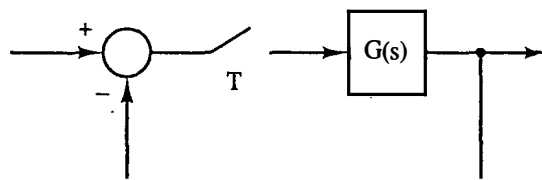


Figure P7-13 System for Problem 7-13.

7-14. For the system of Problem 7-6 and Figure P7-6:

- Plot the z -plane root locus.
- Plot the w -plane root locus.
- Determine the range of K for stability using the results of part (a).
- Determine the range of K for stability using the results of part (b).
- Verify $G(w)$ and the w -plane root locus by digital computation.

7-15. For the chamber temperature control system of Problem 7-9 and Figure P7-9:

- Plot the z -plane root locus.
- Plot the w -plane root locus.
- Determine the range of K for stability using the results of part (a).
- Determine the range of K for stability using the results of part (b).
- Verify $G(w)$ and the w -plane root locus by digital computation.

- 7-16. For the robot arm joint control system of Problem 7-10 and Figure P7-10:
- Plot the z -plane root locus.
 - Plot the w -plane root locus.
 - Determine to the range of K for stability using the results of part (a).
 - Determine to the range of K for stability using the results of part (b).
 - Verify $G(w)$ and the w -plane root locus by digital computation.
- 7-17. For the antenna control system of Problem 7-11 and Figure P7-11:
- Plot the z -plane root locus.
 - Plot the w -plane root locus.
 - Determine to the range of K for stability using the results of part (a).
 - Determine to the range of K for stability using the results of part (b).
 - Verify $G(w)$ by digital computation.
- 7-18. For the satellite control system of Problem 7-12 and Figure P7-12:
- Plot the z -plane root locus.
 - Plot the w -plane root locus.
 - Determine to the range of K for stability using the results of part (a).
 - Determine to the range of K for stability using the results of part (b).
 - Verify $G(w)$ and the w -plane root locus by digital computation.
- 7-19. For the system of Problem 7-6 and Figure P7-6, let $K = 1$.
- Determine the stability of the system.
 - Sketch the Bode diagram, and use this diagram to sketch the Nyquist diagram.
 - If the system is stable, determine the gain and phase margins. If the system is unstable, find the value of K that gives a phase margin of 45° .
 - From the Nyquist diagram, determine the value of $K > 0$ for which the system is marginally stable.
 - Find the frequency ω at which the marginally stable system will oscillate.
 - Verify $G(w)$ by digital computation.
- 7-20. Consider the general bilinear transformation

$$z = \frac{1 + aw}{1 - aw}$$

where a is real and nonzero.

- Show that this function transforms the stability boundary of the z -plane into the imaginary axis in the w -plane.
 - Find the relationship of frequency in the s -plane to frequency in the w -plane.
 - Find the stable region of the w -plane for $a < 0$ and for $a > 0$.
- 7-21. Given the pulse transfer function $G(z)$ of a plant. For $\omega = 2$ rad/s, $G(e^{j\omega T})$ is equal to the complex number $1.3/-25^\circ$. The signal $5 \cos 2t$ is applied to the input (sampler and data hold) of the plant. Find the steady-state sampled output.
- 7-22. For the temperature control system of Problem 7-9 and Figure P7-9, let $K = 1$.
- Determine the stability of the system.
 - Sketch the Bode diagram, and use this diagram to sketch the Nyquist diagram.
 - If the system is stable, determine the gain and phase margins. If the system is unstable, find the value of K that gives a phase margin of 45° .
 - From the Nyquist diagram, determine the value of $K > 0$ for which the system is marginally stable.
 - Find the frequency ω at which the marginally stable system will oscillate.
 - Verify $G(w)$ by digital computation.
- 7-23. For the robot arm joint control system of Problem 7-10 and Figure P7-10, let $K = 1$.
- The frequency response for $G(z)$ was calculated by computer and is given in

TABLE P7-23 FREQUENCY RESPONSE FOR PROBLEM 7-23

ω_w	ω	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	19.97513	26.00	-93.14
0.2	0.200	9.95054	19.95	-96.28
0.3	0.300	6.59317	16.38	-99.38
0.4	0.399	4.90325	13.80	-102.45
0.5	0.499	3.88102	11.77	-105.46
0.6	0.599	3.19331	10.08	-108.41
0.7	0.699	2.69741	8.61	-111.28
0.8	0.799	2.32198	7.31	-114.08
0.9	0.899	2.02741	6.13	-116.78
1.0	0.999	1.78990	5.05	-119.40
2.0	1.993	0.70945	-2.98	-140.61
3.0	2.977	0.37308	-8.56	-154.64
4.0	3.947	0.22743	-12.86	-164.43
5.0	4.899	0.15270	-16.32	-171.82
6.0	5.829	0.10973	-19.19	-177.74
7.0	6.733	0.08291	-21.62	-182.72
8.0	7.610	0.06511	-23.72	-187.04
9.0	8.457	0.05270	-25.56	-190.88
10.0	9.273	0.04372	-27.18	-194.33
20.0	15.708	0.01403	-37.05	-217.40
30.0	19.656	0.00798	-41.96	-229.64
40.0	22.143	0.00558	-45.07	-236.77

Table P7-23. Sketch the Nyquist diagram for the open-loop function $G(z)H_k$, with $H_k = 0.07 \Rightarrow -23.1$ dB.

- (b) Determine the stability of the system.
 - (c) If the system is stable, determine the gain and phase margins. If the system is unstable, find the value of K that gives a phase margin of 45° .
 - (d) From the Nyquist diagram, determine the value of $K > 0$ for which the system is marginally stable.
 - (e) Use the frequency response to find the frequency ω at which the marginally stable system will oscillate.
- 7-24. For the antenna control system of Problem 7-11 and Figure P7-11, let $K = 1$.
- (a) The frequency response for $G(z)$ was calculated by computer and is given in Table P7-24. Sketch the Nyquist diagram for the open-loop function $G(z)H_k$, with $H_k = 0.04 \Rightarrow -7.96$ dB.
 - (b) Determine the stability of the system.
 - (c) If the system is stable, determine the gain and phase margins. If the system is unstable, find the value of K that gives a phase margin of 45° .
 - (d) From the Nyquist diagram, determine the value of $K > 0$ for which the system is marginally stable.
 - (e) Find the frequency ω at which the marginally stable system will oscillate.
- 7-25. For the satellite control system of Problem 7-12, the frequency response for $G(z)$ was calculated by computer and is given in Table P7-25.
- (a) Sketch the Nyquist diagram for the open-loop function $G(z)H$, with $H = 0.02 \Rightarrow -34.0$ dB.
 - (b) Use the results in part (a) to determine the range of K for stability.

TABLE P7-24 FREQUENCY RESPONSE FOR
PROBLEM 7-24

ω_w	ω	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	33.32874	0.00	-91.09
0.2	0.200	16.65748	0.00	-92.19
0.3	0.300	11.09735	0.00	-93.29
0.4	0.400	8.31502	0.01	-94.38
0.5	0.500	6.64381	0.02	-95.47
0.6	0.600	5.52819	0.03	-96.56
0.7	0.699	4.73007	0.05	-97.65
0.8	0.799	4.13040	0.06	-98.73
0.9	0.899	3.66305	0.08	-99.81
1.0	0.999	3.28834	0.09	-100.89
2.0	1.998	1.58193	0.26	-111.28
3.0	2.994	0.99511	0.08	-120.81
4.0	3.986	0.69524	-0.95	-129.31
5.0	4.974	0.51456	-2.88	-136.78
6.0	5.955	0.39576	-5.22	-143.31
7.0	6.929	0.31327	-7.57	-149.03
8.0	7.895	0.25375	-9.75	-154.07
9.0	8.852	0.20951	-11.73	-158.54
10.0	9.799	0.17582	-13.52	-162.54
20.0	18.546	0.05320	-25.01	-188.55
30.0	25.740	0.02706	-31.13	-203.49
40.0	31.416	0.01738	-35.07	-213.67

TABLE P7-25 FREQUENCY RESPONSE FOR
PROBLEM 7-25

ω_w	ω	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	1000.01200	60.00	-180.28
0.2	0.200	250.01250	47.95	-180.57
0.3	0.300	111.12360	40.91	-180.85
0.4	0.399	62.51249	35.91	-181.14
0.5	0.499	40.01249	32.04	-181.43
0.6	0.599	27.79027	28.87	-181.71
0.7	0.699	20.42065	26.20	-182.00
0.8	0.799	15.63749	23.88	-182.29
0.9	0.899	12.35817	21.83	-182.57
1.0	0.999	10.01249	20.01	-182.86
2.0	1.993	2.51247	8.00	-185.71
3.0	2.977	1.12354	1.01	-188.53
4.0	3.947	0.63738	-3.91	-191.30
5.0	4.899	0.41231	-7.69	-194.03
6.0	5.829	0.29001	-10.75	-196.69
7.0	6.733	0.21622	-13.30	-199.29
8.0	7.610	0.16829	-15.47	-201.80
9.0	8.457	0.13538	-17.36	-204.22
10.0	9.273	0.11180	-19.03	-206.56
20.0	15.708	0.03536	-29.03	-225.00
30.0	19.656	0.02003	-33.96	-236.30
40.0	22.143	0.01398	-37.09	-243.43

Digital Controller Design

8.1 INTRODUCTION

In the preceding chapters we have been concerned primarily with analysis. We have assumed that the control system was given, and we analyzed the system to determine stability, stability margins, time response, frequency response, and so on. Some simple design problems were considered: for example, the determination of gains required to meet steady-state error specifications.

In this chapter we consider the total design problem: How do we design a digital controller transfer function (or difference equation) that will satisfy design specifications for a given control system? We will investigate the classical design techniques of frequency response and root locus. First, phase-lag and phase-lead controllers are considered. Then a particular type of lag-lead controller, called a proportional-plus-integral-plus-derivative (PID) controller, is developed. Finally, design by root-locus procedures is introduced. Further design techniques, which are based on the state-variable model of the plant, are developed in Chapters 9 and 10.

The preceding paragraph requires additional comment. All numerical design procedures are based on an inexact model of the physical system. Hence numerical design simply gets us to the point that we can experiment with the physical system, or with an accurate simulation that includes the system nonlinearities, time-varying components, and so on. Design is generally too complex if the accurate simulation model, sometimes called a truth model in optimal filter design, is used. Thus, in one sense, any type of numerical design procedure is trial and error, with the final form and coefficients of the controller determined by several iterations of first numerical



Modern military aircraft utilize many digital control systems. (Courtesy of McDonnell-Douglas Corporation.)

design and then experimentation with either the physical system or an accurate simulation.

8.2 CONTROL SYSTEM SPECIFICATIONS

The design of a control system involves the changing of system parameters and/or the addition of subsystems (called compensators) to achieve certain desired system characteristics. The desired characteristics, or performance specifications, generally relate to steady-state accuracy, transient response, relative stability, sensitivity to change in system parameters, and disturbance rejection. These performance specifications will now be discussed [1].

Steady-State Accuracy

Since steady-state accuracy was discussed in detail in Section 6.5, only a brief review will be given here. In Section 6.5, it was shown that steady-state accuracy is increased if poles at $z = 1$ are added to the open-loop function, and/or if the open-loop gain is increased. However, added poles at $z = 1$ in the open-loop function introduce phase lag into the open-loop frequency response, resulting in reduced stability margins. Thus stability problems may ensue. In addition, an increase in the open-loop gain generally results in stability problems, as was seen in Chapter 7. Thus a control

system design is usually a trade-off between steady-state accuracy and acceptable relative stability (acceptable stability margins).

Transient Response

We define a physical system that has *two dominant poles* as one that can be modeled with reasonable accuracy by a second-order transfer function. Figure 8-1a illustrates a typical step response for a system that has two dominant complex poles. Typical performance criteria are rise time t_r , peak overshoot M_p , time-to-peak overshoot t_p , and settling time t_s . Rise time in this figure is the time required for the response to rise from 10 percent to 90 percent of the final value. However, other definitions are also used for rise time, but all are similar. Settling time t_s is defined as the time required for the response to settle to within a certain percent of the final value. Typical percentage values used are 2 percent and 5 percent.

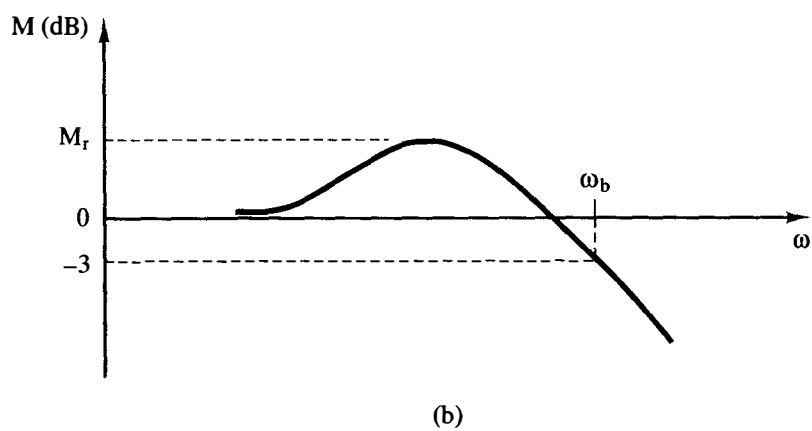
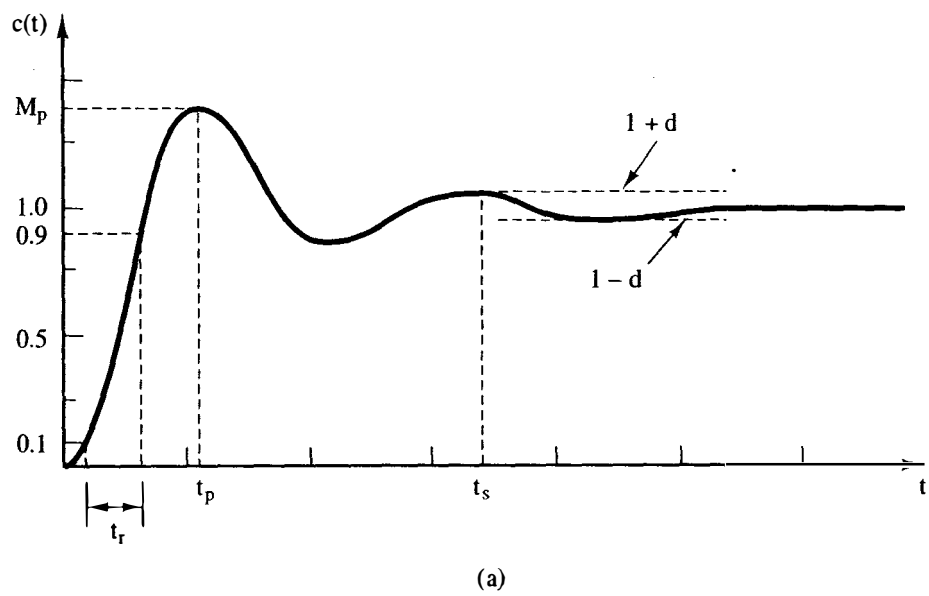


Figure 8-1

For a given system, the time response is uniquely related to the closed-loop frequency response. However, except for first- and second-order systems, the exact relationship is complex and is generally not used. As indicated in Section 7.10, a typical closed-loop frequency response is as shown in Figure 8-1b, where only the magnitude is shown. In this figure, M_r is the resonant peak value of the frequency response, and as was implied in Example 7.13, a larger resonant peak value indicates a larger peak overshoot M_p in the step response. For example, a control-system specification sometimes used is to limit M_r to 2 dB in order to limit M_p to a reasonable value.

Consider the standard second-order LTI analog system with the transfer function

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8-1)$$

which has the unit-step response [1]

$$c(t) = 1 - \frac{1}{\beta} e^{-\zeta\omega_n t} \sin(\beta\omega_n t + \theta)$$

where $\beta = \sqrt{1 - \zeta^2}$ and $\theta = \tan^{-1}(\beta/\zeta)$. Shown in Figure 8-2 are the relationships

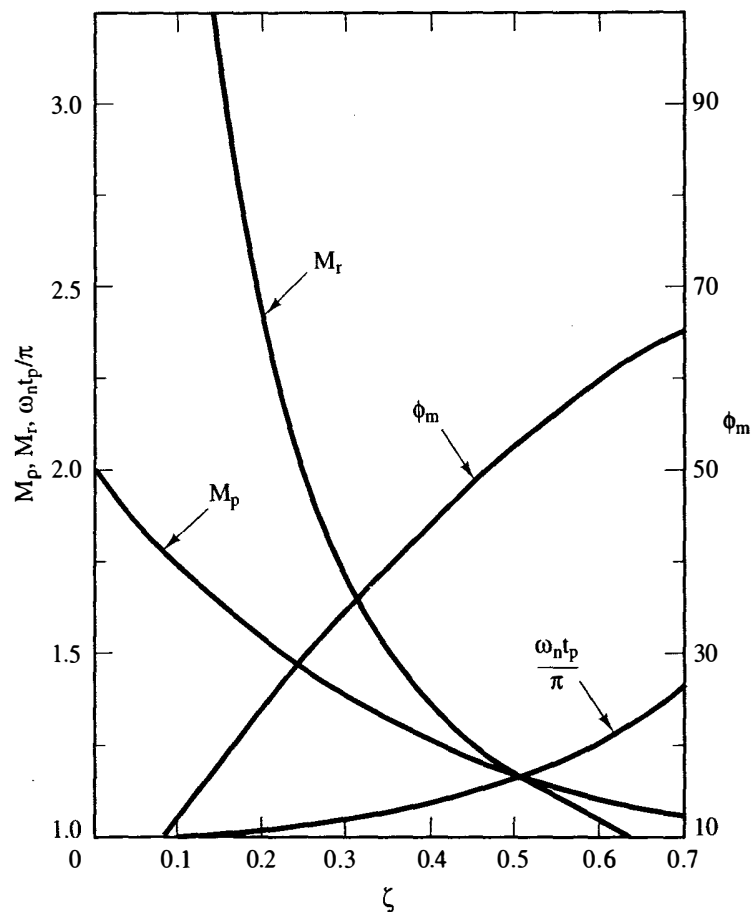


Figure 8-2 Characteristics of a second-order system.

of the parameters M_p , M_r , and t_p of Figure 8-1 to the damping ratio ζ for this system. These plots are obtained from the equations [1]

$$\begin{aligned} M_p &= 1 + e^{-\zeta\pi/\sqrt{1-\zeta^2}} \\ M_r &= \frac{1}{2\zeta\sqrt{1-\zeta^2}} \\ \frac{\omega_n t_p}{\pi} &= \frac{1}{\sqrt{1-\zeta^2}} \end{aligned} \quad (8-2)$$

The relationships apply only approximately to second-order complex poles of higher-order discrete systems (see Section 6.4). However, we see the importance of the damping ratio ζ of complex poles. The fourth curve in Figure 8-2 will be discussed in the next section.

In the damped sinusoidal response term of the second-order system response $c(t)$ just given, the time constant is seen to be $\tau = 1/(\zeta\omega_n)$. This time constant determines the settling time t_s of Figure 8-1a. For example, for the response $c(t)$ to settle to within 2 percent of the final value,

$$e^{-t_s/\tau} = 0.02$$

or $t_s = 3.9\tau$. Hence the system response settles out in approximately four time constants.

The transient response is also related to the system bandwidth, shown as ω_b in Figure 8-1b. For a system the product of rise time and bandwidth (i.e., the product $t_r \omega_b$), is approximately constant [1,2]. Thus, to decrease rise time and increase speed of response, it is necessary that the system bandwidth be increased. However, if significant high-frequency noise sources are present in the system, a larger bandwidth will increase the system response to these noise sources. In this case, a trade-off must be made between a fast rise time and an acceptable noise response.

Relative Stability

In Chapter 7 the relative stability measurements, gain margin and phase margin, were introduced. These margins are an approximate indication of the closeness of the Nyquist diagram (open-loop frequency response) to the -1 point. As was shown in Section 7.10, the closeness of the open-loop frequency response to the -1 point in the complex plane determines the resonant peak value M_r of the closed-loop frequency response (see Figure 8-1b). And M_r is related, in an approximate sense, to the peak overshoot M_p (Figure 8-1) in the step response. Thus, in an approximate sense, the stability margins are related to peak overshoot M_p . Consider the closed-loop function of (8-1) for which the open-loop function $G_p(s)$ is given by

$$G_p(s) = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)} \Rightarrow T(s) = \frac{G_p(s)}{1 + G_p(s)}$$

For this system the phase margin ϕ_m and the peak overshoot M_p are directly related, as shown in Figure 8-2. The curve plotted in Figure 8-2 is the relationship [1]

$$\phi_m = \tan^{-1} \left[\frac{2\zeta}{(\sqrt{4\zeta^4 + 1} - 2\zeta^2)^{1/2}} \right]$$

This equation is often approximated by (see Problem 8-1)

$$\phi_m \approx 100\zeta$$

Sensitivity

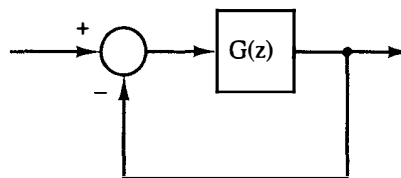
Generally, any control system will contain parameters that change with temperature, humidity, altitude, age, and so on. However, we prefer that the control system characteristics not vary as these parameters vary. Of course, the system characteristics are a function of the system parameters, but in some cases the sensitivity of system characteristics to parameter variations can be reduced. A simple case will now be discussed.

Consider the discrete system of Figure 8-3a. For this system the closed-loop transfer function $T(z)$ is given by

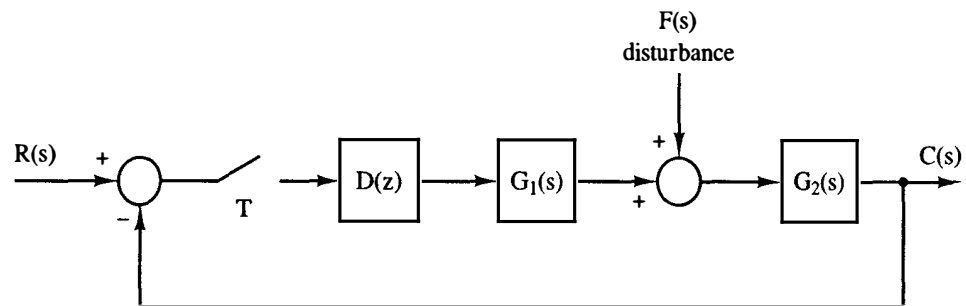
$$T(z) = \frac{G(z)}{1 + G(z)}$$

Sensitivity to a parameter, a , is normally defined as a measure of the percentage change in $T(z)$ to a percentage change in the parameter, a . One such definition is

$$\text{sensitivity} \approx \frac{\Delta T/T}{\Delta a/a} = \frac{\Delta T}{\Delta a} \frac{a}{T} \quad (8-3)$$



(a)



(b)

Figure 8-3 Discrete control systems.

where ΔT is the variation in T caused by Δa , the variation in parameter a . If the limit of (8-3) is taken as $\Delta a \rightarrow 0$, we get the usual definition of *sensitivity*; that is,

$$S_a^T = \frac{\partial T}{\partial a} \frac{a}{T} \quad (8-4)$$

We will now find the sensitivity of T with respect to G for the system of Figure 8-3a.

$$\begin{aligned} S_G^T &= \frac{\partial T}{\partial G} \cdot \frac{G}{T} = \frac{1 + G(z) - G(z)}{[1 + G(z)]^2} \cdot \frac{G(z)}{G(z)/[1 + G(z)]} \\ &= \frac{1}{1 + G(z)} \end{aligned}$$

At the frequency ω , we let $z = e^{j\omega T}$, and

$$S_G^T = \frac{1}{1 + G(e^{j\omega T})} \quad (8-5)$$

For this sensitivity to be small within the system bandwidth, we require that $G(e^{j\omega T}) \gg 1$. Thus we can reduce the sensitivity of T to G by increasing the open-loop gain. But as noted before, increasing the open-loop gain can cause stability problems. Thus, once again, in design we are faced with trade-offs.

Consider now that $G(z)$ is a function of the parameter a . Then we can express (8-4) as

$$S_a^T = \frac{\partial T}{\partial a} \frac{a}{T} = \frac{\partial T}{\partial G} \frac{\partial G}{\partial a} \frac{a}{T}$$

Thus

$$\begin{aligned} S_a^T &= \frac{1 + G(z) - G(z)}{[1 + G(z)]^2} \frac{a}{G(z)/[1 + G(z)]} \frac{\partial G(z)}{\partial a} \\ &= \frac{a \frac{\partial G(z)}{\partial a}}{G(z)[1 + G(z)]} \end{aligned} \quad (8-6)$$

Then, as in (8-5), to reduce the sensitivity we must increase loop gain.

Disturbance Rejection

A control system will generally have inputs other than the one to be used to control the system output. An example is shown in Figure 8-3b. In this system $F(s)$ is a disturbance. Since $R(s)$ is the control input, we design the system such that $c(t)$ is approximately equal to $r(t)$. If $F(s)$ is zero, then

$$C(z) = \frac{D(z)\overline{G_1 G_2(z)}}{1 + D(z)\overline{G_1 G_2(z)}} R(z)$$

Hence, in terms of the frequency response, we require that

$$D(e^{j\omega T}) \overline{G_1 G_2}(e^{j\omega T}) \gg 1$$

over the desired system bandwidth. Then

$$C(e^{j\omega T}) \approx R(e^{j\omega T})$$

If we consider only the disturbance input in Figure 8-3b, then

$$C(z) = \frac{\overline{G_2 F}(z)}{1 + D(z) \overline{G_1 G_2}(z)}$$

Hence, over the desired system bandwidth,

$$C(e^{j\omega T}) \approx \frac{\overline{G_2 F}(e^{j\omega T})}{D(e^{j\omega T}) \overline{G_1 G_2}(e^{j\omega T})}$$

Since the denominator of this expression is large, the disturbance response will be small, provided that the numerator is not large. Therefore, we generally have good disturbance rejection in a system provided that we have a high loop gain, and provided that the high loop gain does not occur in the direct path between the disturbance input and the system output [$G_2(s)$ in Figure 8-3b].

Control Effort

Another criterion that must be considered in the design of a control system is the *control effort*. For example, generally in a radar tracking system, an electric motor is used to rotate the radar antenna. Any physical motor will have a maximum torque that can be developed. If we call this control effort (the torque) $u(t)$, then $|u(t)|$ will be bounded. One procedure for including this constraint in the design of the system is to first design without considering the maximum torque available. Next the designed system is simulated under worst-case conditions to determine the maximum torque required. Then a physical motor is chosen that can produce this value of torque. This design procedure may be iterative, since the chosen motor may not have the model assumed in the initial design.

Another example of constraints on control effort is the maximum energy that may be available over a period of time. This constraint is usually stated as

$$\int_0^{t_f} |u^2(t)| dt \leq M$$

For example, certain types of attitude controllers for satellites have limited energy available. The topic of constraints on the control effort is covered in the design of certain types of optimal control systems in Chapter 10.

8.3 COMPENSATION

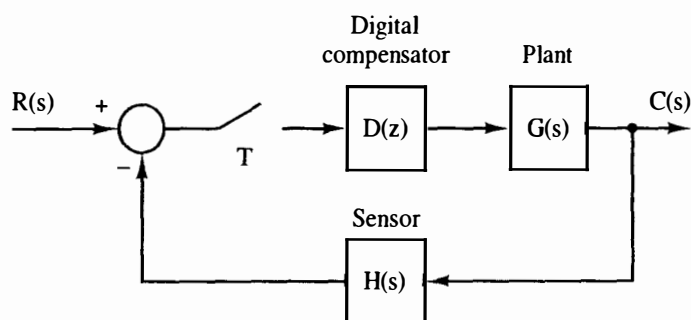
In this chapter we will, for the most part, limit the discussion to the design of compensators for single-input, single-output systems. A simple system of this type is shown in Figure 8-4a. For this system,

$$\frac{C(z)}{R(z)} = \frac{D(z)G(z)}{1 + D(z)\overline{GH}(z)} \quad (8-7)$$

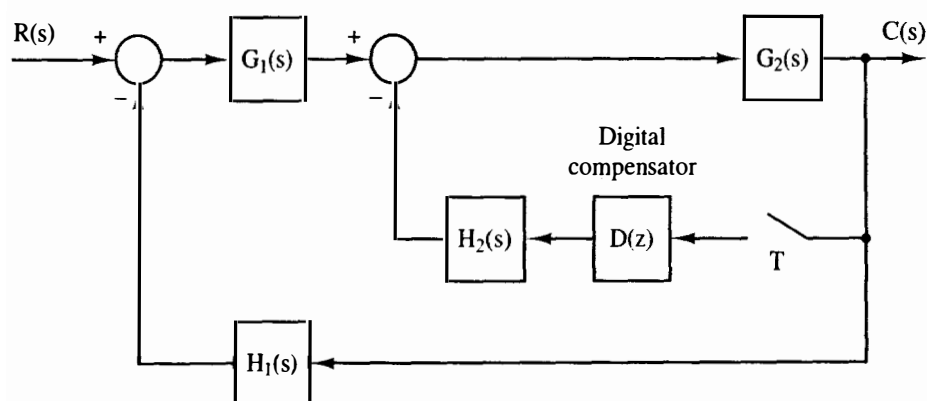
and hence the characteristic equation is

$$1 + D(z)\overline{GH}(z) = 0 \quad (8-8)$$

We call compensation of the type shown in Figure 8-4a *cascade*, or *series*, *compensation*. The effects of this compensation on system characteristics are given by the characteristic equation (8-8).



(a)



(b)

Figure 8-4 Digital control systems.

It is sometimes more feasible to place the compensator within a loop internal to the system. Such a system is illustrated in Figure 8-4b. For this system,

$$C(z) = \frac{\frac{G_1 G_2 R}{1 + G_1 G_2 H_1}}{1 + D(z) \frac{G_2 H_2}{1 + G_1 G_2 H_1}} \quad (8-9)$$

The system characteristic equation is then

$$1 + D(z) \frac{G_2 H_2}{1 + G_1 G_2 H_1} = 0 \quad (8-10)$$

This type of compensation is termed *feedback, parallel, or minor-loop compensation*. For this system, the effects of compensation on system characteristics are given by (8-10).

In the following three sections, basically we will consider compensation by a first-order device. Thus the compensator transfer function can be expressed as

$$D(z) = \frac{K_d(z - z_0)}{z - z_p} \quad (8-11)$$

The design of the compensator in these three sections will be performed in the frequency domain using Bode techniques; thus we will be working in the w -plane. The transformation of $D(z)$ to the w -plane yields $D(w)$; that is,

$$D(w) = D(z) \Big|_{z = [1 + (T/2)w]/[1 - (T/2)w]} \quad (8-12)$$

Thus $D(w)$ is also first order, and we will assume it to be of the form

$$D(w) = a_0 \frac{1 + w/\omega_{w0}}{1 + w/\omega_{wp}} \quad (8-13)$$

where ω_{w0} is the zero location and ω_{wp} is the pole location, in the w -plane. The dc gain of the compensator is found in (8-11) by letting $z = 1$, or in (8-13) by letting $w = 0$. Hence a_0 is the compensator dc gain.

To realize the compensator, the transfer function must be expressed in z , as in (8-11). Then, from (7-8) and (8-13),

$$D(z) = a_0 \frac{1 + \frac{w}{\omega_{w0}}}{1 + \frac{w}{\omega_{wp}}} \Big|_{w = (2/T)[(z-1)/(z+1)]} = a_0 \frac{\omega_{wp}(\omega_{w0} + 2/T)}{\omega_{w0}(\omega_{wp} + 2/T)} \frac{z - \frac{2T - \omega_{w0}}{2T + \omega_{w0}}}{z - \frac{2T - \omega_{wp}}{2T + \omega_{wp}}} \quad (8-14)$$

Hence, in (8-11),

$$K_d = a_0 \frac{\omega_{wp}(\omega_{w0} + 2/T)}{\omega_{w0}(\omega_{wp} + 2/T)}, \quad z_0 = \frac{2T - \omega_{w0}}{2T + \omega_{w0}}, \quad z_p = \frac{2T - \omega_{wp}}{2T + \omega_{wp}} \quad (8-15)$$

The compensator of (8-13) is classified by the location of the zero, ω_{w0} , relative to that of the pole, ω_{wp} . If $\omega_{w0} < \omega_{wp}$, the compensation is called *phase lead*. If $\omega_{w0} > \omega_{wp}$, the compensation is called *phase lag*. The phase-lag compensator will be discussed first.

8.4 PHASE-LAG COMPENSATION

In (8-13), for $\omega_{w0} > \omega_{wp}$, the frequency response of $D(w)$ exhibits a negative phase angle, or phase lag. The frequency response of $D(w)$, as given by a Bode plot, is shown in Figure 8-5. The dc gain is a_0 , and the high-frequency gain is

$$(\text{high-frequency gain})_{\text{dB}} = 20 \log \frac{a_0 \omega_{wp}}{\omega_{w0}} \quad (8-16)$$

The phase characteristic is also shown in Figure 8-5. The maximum phase shift is denoted as ϕ_M , and has a value between 0 and -90° , depending on the ratio ω_{w0}/ω_{wp} .

Design using phase-lag digital compensators will be discussed relative to the system of Figure 8-6. For this system the characteristic equation is given by

$$1 + D(z)G(z) = 0 \quad (8-17)$$

where

$$G(z) = \mathcal{Z} \left[\frac{1 - e^{-Ts}}{s} G_p(s) \right] \quad (8-18)$$

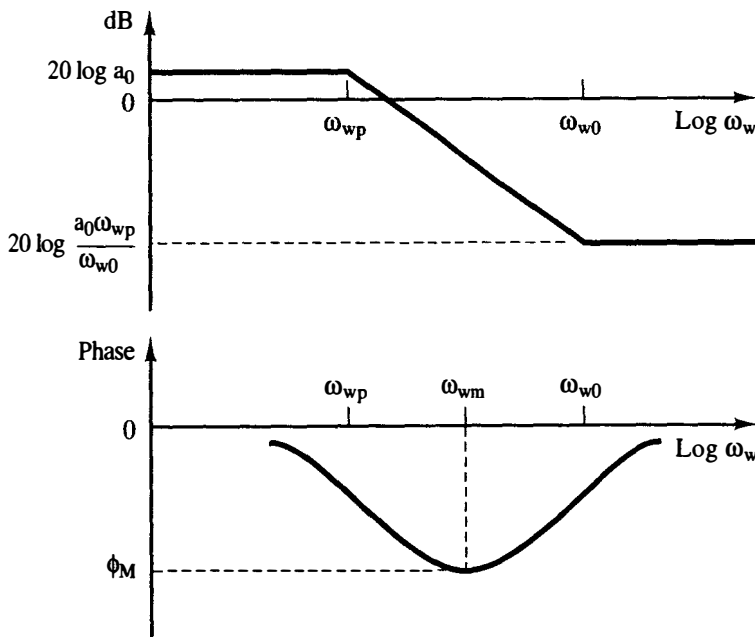


Figure 8-5 Phase-lag digital filter frequency-response characteristics.

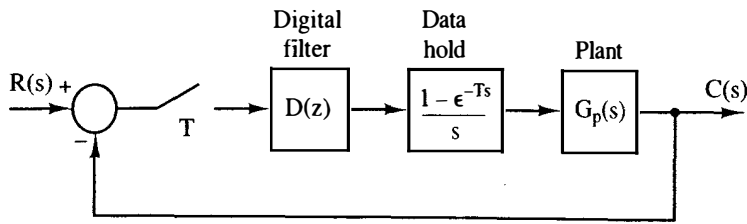


Figure 8-6 Digital control system.

For a system configuration differing from that of Figure 8-6, the characteristic equation is formed as in (8-17), and the frequency response of the transfer function that multiplies $D(z)$, as in (8-17), is calculated. From that point, the design procedure follows that given below.

As is seen from Figure 8-5, phase-lag filters reduce the high-frequency gain relative to the low-frequency gain and introduce phase lag. Since, in general, phase lag tends to destabilize a system (rotates the Nyquist diagram toward the -1 point) the break frequencies, ω_{wp} and ω_{w0} , must be chosen such that the phase lag does not occur in the vicinity of the 180° crossover point of the plant frequency response $G(j\omega_w)$, where

$$G(w) = z \left[\frac{1 - e^{-Ts}}{s} G_p(s) \right]_{z = [1 + (T/2)w]/[1 - (T/2)w]} \quad (8-19)$$

for the system of Figure 8-6. However, for stability purposes, it is necessary that the filter introduce the reduced gain in the vicinity of 180° crossover. Thus, both ω_{wp} and ω_{w0} must be much smaller than the 180° crossover frequency. Figure 8-7 illustrates design by phase-lag compensation, where the compensator dc gain is unity.

Note that in Figure 8-7, both the system gain margin and the system phase margin ϕ_m have been increased by the compensation, increasing relative stability. In addition, the low-frequency gain has not been reduced, and thus steady-state errors and low-frequency sensitivity have not been increased to attain the improved relative stability. The bandwidth has been decreased, which will generally result in a slower system time response.

Suppose that in Figure 8-7, we keep both ω_{w0} and the product $a_0 \omega_{wp}$ constant, and increase a_0 while decreasing ω_{wp} . The high-frequency gain of the compensator remains constant, from (8-16). However, since we are increasing a_0 , the system open-loop low-frequency gain increases. Hence the closed-loop low-frequency gain approaches unity and the steady-state response is improved. However, for a given system, the increase in phase lag added to system phase characteristics may push the phase characteristic below the -180° line (see Figure 8-7). Then we have a *conditionally stable system*, that is, one that can be forced unstable by reducing gain. If the system contains a saturation nonlinearity, large signals into this nonlinearity reduce its effective gain [1,3]. Thus a phase-lag compensated system may exhibit instabilities for large signals (nonlinear operation).

A technique for determining ω_{wp} and ω_{w0} to yield a desired phase margin will now be given. It is assumed that the compensator dc gain, a_0 , is determined from

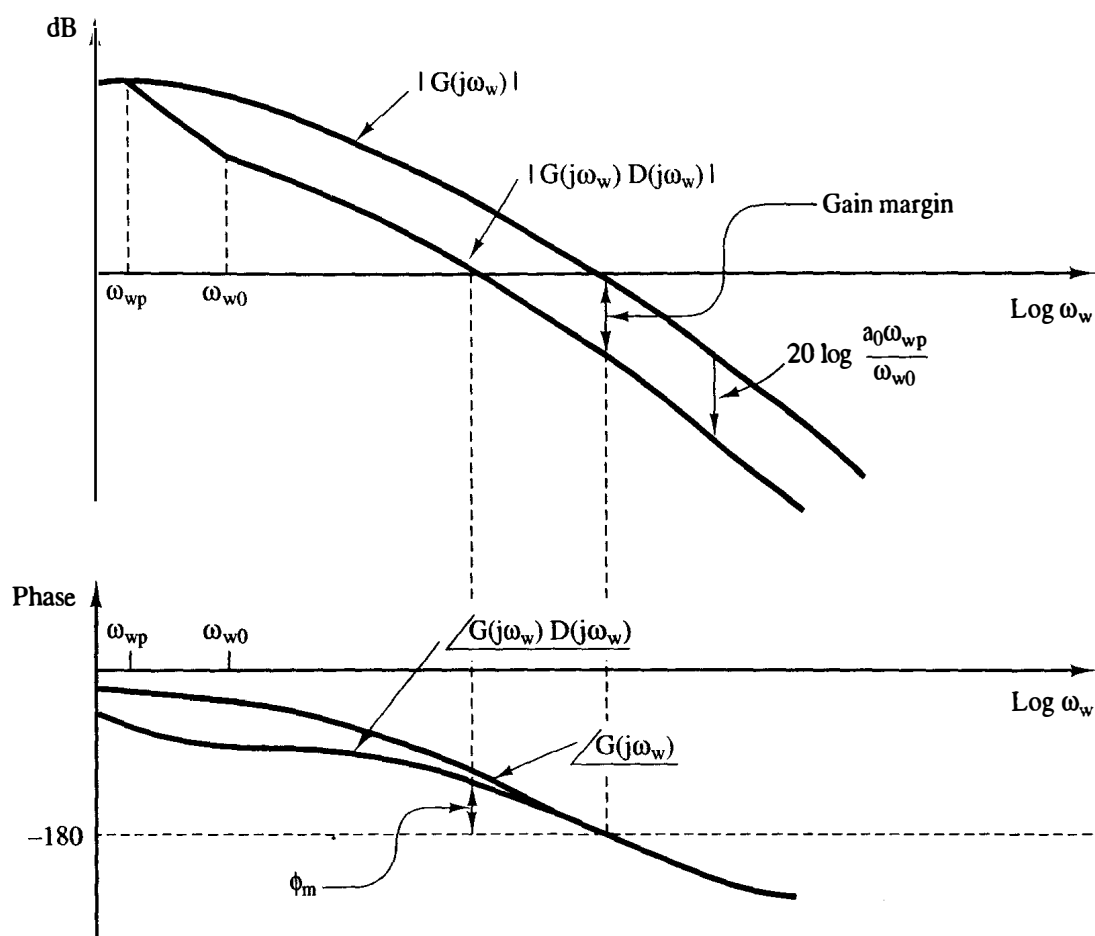


Figure 8-7 Design using phase-lag compensation.

the system specifications, and that the phase margin of ϕ_m is desired. Refer to Figure 8-7. The design steps are:

1. Determine the frequency, ω_{w1} , at which the phase angle of $G(j\omega_w)$ is approximately $(-180^\circ + \phi_m + 5^\circ)$. The phase margin of the compensated system will occur at approximately this frequency.
2. Choose

$$\omega_{w0} = 0.1\omega_{w1} \quad (8-20)$$

to ensure that little phase lag is introduced at ω_{w1} . Actually, the compensator will introduce approximately 5° phase lag, which has been accounted for in step 1

$$\left| \frac{a_0 \omega_{wp}}{\omega_{w0}} G(j\omega_{w1}) \right| = 1 \Rightarrow \frac{a_0 \omega_{wp}}{\omega_{w0}} = \frac{1}{|G(j\omega_{w1})|}$$

(See Figure 8-5.) Solving the last two equations for ω_{wp} yields

$$\omega_{wp} = \frac{0.1\omega_{w1}}{a_0|G(j\omega_{w1})|} \quad (8-21)$$

The design is now complete, since the compensator dc gain, pole, and zero are known.

Once a_0 , ω_{wp} , and ω_{w0} are known, $D(z)$ is obtained from (8-14). For the case that the sensor transfer function $H(s)$ is not unity, replace $G(j\omega_{w1})$ with $\overline{GH}(j\omega_{w1})$ in step 1 and (8-21).

Example 8.1

We will consider the design of a servomotor system as described in Section 1.5. Suppose that the servo is to control the horizontal (azimuth) angle for pointing a radar antenna. Then, in the closed-loop system of Figure 8-6, $c(t)$ is the azimuth angle of the antenna, and $r(t)$ is the commanded, or desired, azimuth angle. The plant transfer function derived in Section 1.5 is second order; however, we will assume that the armature inductance cannot be neglected, resulting in a third-order transfer function. Then suppose that the parameters of the plant are such that

$$G_p(s) = \frac{1}{s(s+1)(0.5s+1)}$$

Since the fastest time constant is 0.5 s, we will choose T to be one-tenth that value (a rule of thumb often used), or $T = 0.05$ s. Then

$$\begin{aligned} G(z) &= \frac{z-1}{z} \mathcal{Z} \left[\frac{1}{s^2(s+1)(0.5s+1)} \right] \\ &= \frac{z-1}{z} \mathcal{Z} \left[\frac{1}{s^2} + \frac{-1.5}{s} + \frac{2}{s+1} + \frac{-0.5}{s+2} \right] \\ &= \frac{z-1}{z} \left[\frac{0.005z}{(z-1)^2} - \frac{1.5z}{z-1} + \frac{2z}{z-0.9512} - \frac{0.5z}{z-0.9048} \right] \end{aligned}$$

The frequency response of this system was calculated by computer and is given in Table 8-1 and plotted in Figure 8-8. Suppose that it is desired to design a unity dc gain phase-lag compensator ($a_0 = 1$) to achieve a phase margin of 55° . Then, using the foregoing procedure, we see that the frequency ω_{w1} occurs where the phase of $G(j\omega_w)$ is $(-180^\circ + 55^\circ + 5^\circ) = -120^\circ$, or $\omega_{w1} \approx 0.36$. At this frequency, $|G(j\omega_1)| \approx 2.57$. Then, from (8-20),

$$\omega_{w0} = 0.1\omega_{w1} = 0.036$$

and from (8-21),

$$\omega_{wp} = \frac{0.1\omega_{w1}}{a_0|G(j\omega_{w1})|} = \frac{0.036}{(1)(2.57)} = 0.0140$$

Then $D(w) = (1 + w/0.036)/(1 + w/0.0140)$, and from (8-14),

$$D(z) = \frac{0.3891(z - 0.998202)}{(z - 0.999300)} = \frac{0.3891z - 0.38840}{z - 0.999300}$$

TABLE 8-1 FREQUENCY RESPONSE OF THE PLANT
IN EXAMPLE 8.1

ω	ω_w	$ G(\epsilon^{j\omega T}) $	$ G(\epsilon^{j\omega T}) _{\text{dB}}$	$\angle G(\epsilon^{j\omega T})$
0.010	0.010	100.0	40.0	-90.9
0.050	0.050	19.97	26.0	-94.4
0.100	0.100	9.94	19.9	-98.7
0.200	0.200	4.88	13.8	-107.3
0.300	0.300	3.16	9.99	-115.6
0.360	0.360	2.57	8.21	-120.5
0.400	0.400	2.28	7.15	-123.7
0.500	0.500	1.74	4.79	-131.3
0.600	0.600	1.37	2.73	-138.5
0.700	0.700	1.105	0.87	-145.3
0.800	0.800	0.9064	-0.85	-151.6
0.900	0.900	0.7533	-2.46	-157.5
1.000	1.000	0.6330	-3.97	-163.0
1.200	1.200	0.4576	-6.79	-172.9
1.370	1.371	0.3550	-8.99	-180.3
1.500	1.501	0.2950	-10.6	-185.4
2.000	2.001	0.1584	-16.0	-201.4
3.000	3.006	0.0590	-24.6	-222.3
5.000	5.026	0.0151	-36.7	-244.3

Calculation of the system open-loop frequency response shows that this compensator results in a gain margin of approximately 16 dB and a phase margin of approximately 55°.

With phase-lag compensation numerical problems may occur in the realization of the filter coefficients. To illustrate this point, suppose that a microprocessor is used to implement the digital controller. Suppose, in addition, that filter coefficients are realized by a binary word that employs 8 bits to the right of the binary point. Then the fractional part of the coefficient can be represented as [4]

$$\text{fraction} = b_7 * \frac{1}{2} + b_6 * \frac{1}{4} + b_5 * \frac{1}{8} + \cdots + b_0 * \frac{1}{2^8}$$

where b_i is the i th bit, and has a value of either zero or 1. For example, the binary number

$$(0.11000001)_2 = \left(\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^8} \right)_{10} = (0.75390625)_{10}$$

The maximum value that the fraction can assume is $[1 - 1/(2)^8]$, or 0.99609375. Note, in Example 8.1, that a denominator coefficient of 0.999300 is required, but a value of 0.99609375 will be implemented (b_7 to b_0 are all equal to 1). The numerator coefficients, when converted by standard decimal-to-binary conversion algorithms [4], become

$$(0.3891)_{10} \Rightarrow (0.01100011)_2 = 0.38671875$$

$$(0.38840)_{10} \Rightarrow (0.01100011)_2 = 0.38671875$$

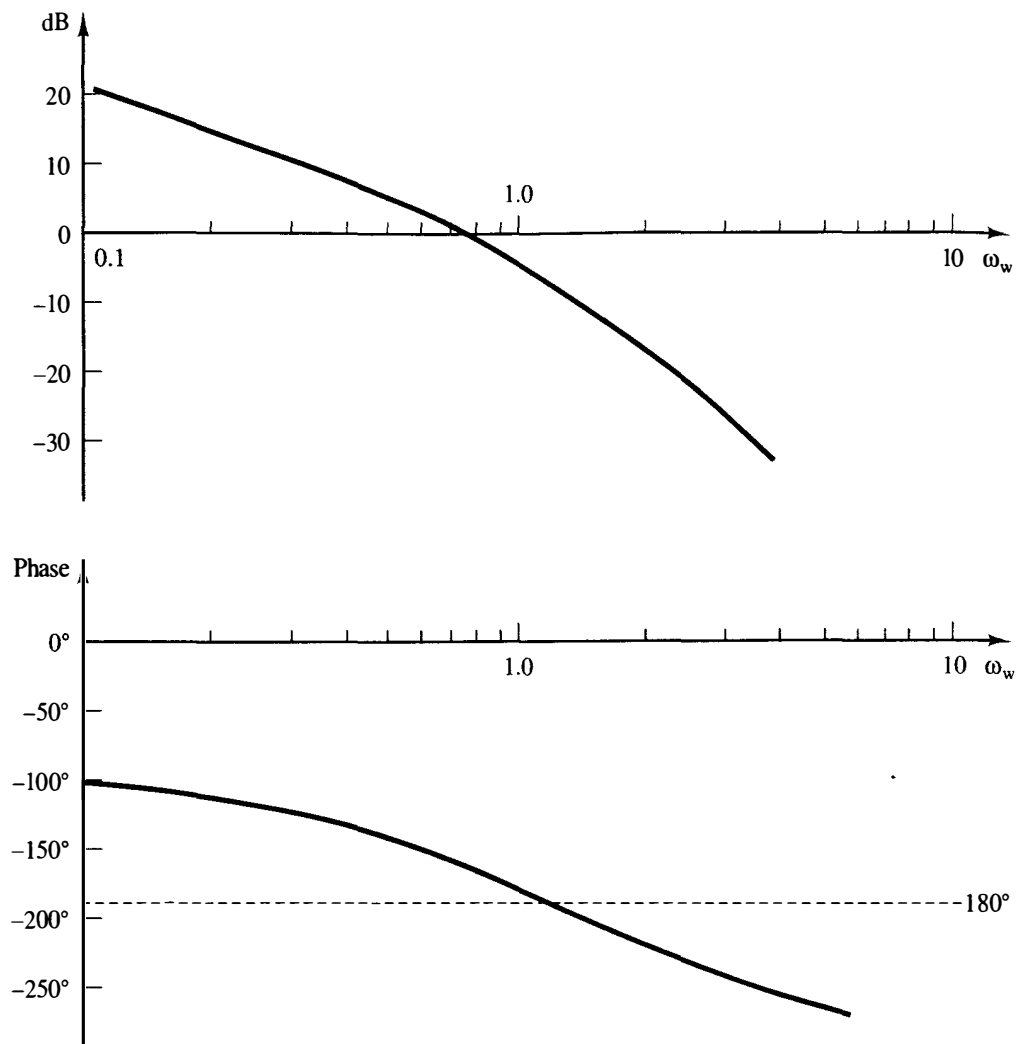


Figure 8-8 Frequency response for the system of Example 8.1.

Thus the compensator zero has been shifted to $z = 1$, and the digital filter that is implemented has the transfer function

$$D(z) = \frac{0.38671875z - 0.38671875}{z - 0.99609375}$$

Shown in Figure 8-9 are the frequency responses of the designed filter and the implemented filter, and the effects of coefficient quantization are evident. The resultant system stability margins, when the implemented filter is used, are: phase margin 70° (designed value 55°), and gain margin 18 dB (designed value 16 dB). However, the implemented filter has a dc gain of zero; thus the system will not respond correctly to a constant input. Hence more bits must be used to represent the filter coefficients. Coefficient quantization effects are investigated in detail in Chapter 14.

We can view the coefficient quantization problem as one that results from the choice of the sample period T . We place digital filters in a physical system in order

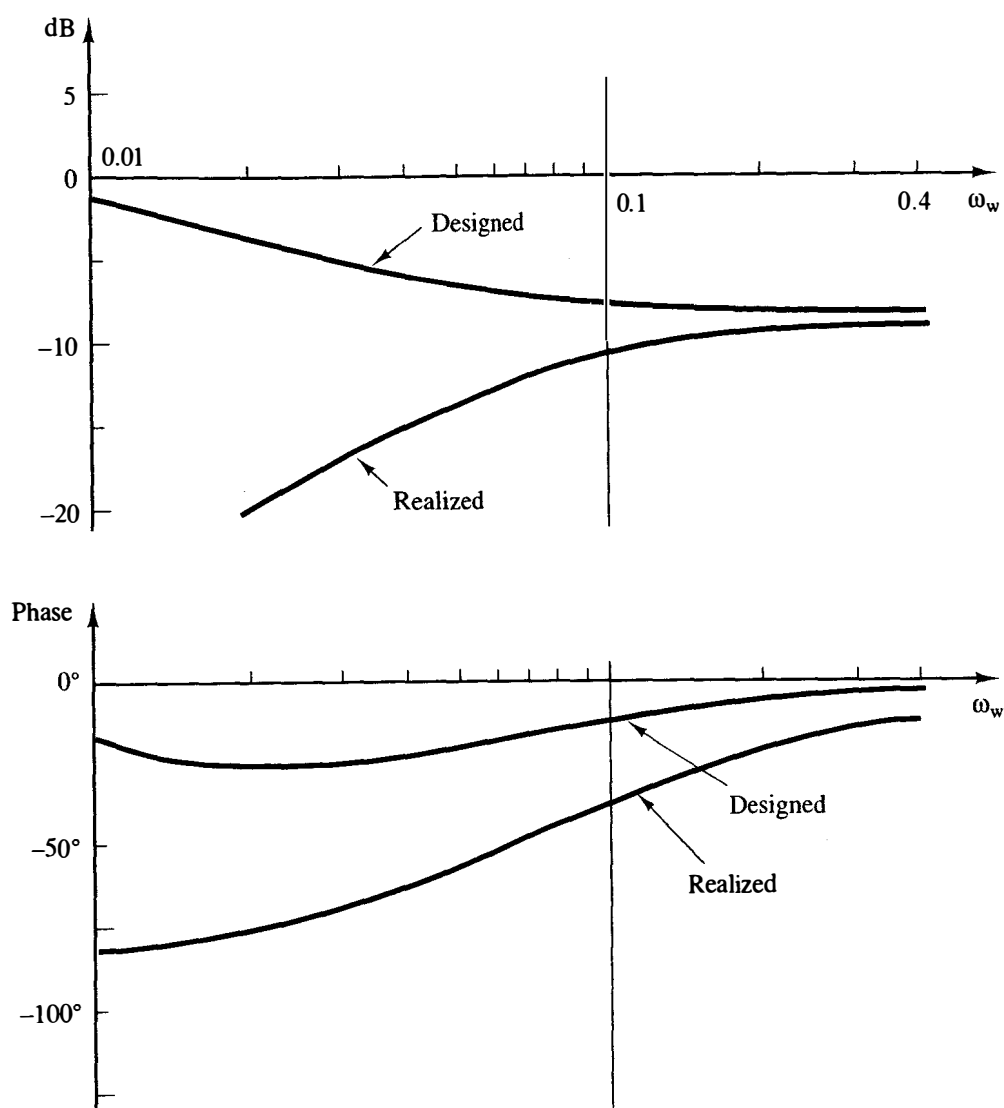


Figure 8-9 Frequency responses of designed and of realized digital controllers.

to change its real (s -plane) frequency response, and we want this change to occur over a certain real frequency (ω) range. The choice of T places this frequency range on a certain part of the unit circle in the z -plane, since $z = e^{j\omega T} = 1/\omega T$. Thus since phase-lag filtering occurs for ω small, the choice of T small requires that the filtering occur in the vicinity of the $z = 1$ point. Thus the phase-lag pole and zero will occur close to $z = 1$, and thus close to each other. If T can be chosen to be a larger value, the phase-lag pole and zero will move away from the $z = 1$ point (and each other), and the numerical accuracy required for the filter coefficients will not be as great.

8.5 PHASE-LEAD COMPENSATION

Phase-lead compensation will now be discussed. For a phase-lead compensator, in (8-13) $\omega_{w0} < \omega_{wp}$, and the compensator frequency response is as shown in Figure

8-10. The maximum phase shift, θ_M , occurs at a frequency ω_{wm} , where ω_{wm} is the geometric mean of ω_{w0} and ω_{wp} , that is,

$$\omega_{wm} = \sqrt{\omega_{w0} \omega_{wp}} \quad (8-22)$$

A plot of θ_M versus the ratio ω_{wp}/ω_{w0} is given in Figure 8-11. This plot was obtained through the following development. We can express (8-13) as

$$D(j\omega_w) = |D(j\omega_w)|e^{j\theta} = a_0 \left[\frac{1 + j(\omega_w/\omega_{wp})}{1 + j(\omega_w/\omega_{w0})} \right] \quad (8-23)$$

Then

$$\tan \theta = \tan \left[\tan^{-1} \frac{\omega_w}{\omega_{w0}} - \tan^{-1} \frac{\omega_w}{\omega_{wp}} \right] = \tan(\alpha - \beta) \quad (8-24)$$

Thus

$$\tan \theta = \frac{\tan \alpha - \tan \beta}{1 + \tan \alpha \tan \beta} = \frac{\omega_w/\omega_{w0} - \omega_w/\omega_{wp}}{1 + \omega_w^2/\omega_{w0} \omega_{wp}} \quad (8-25)$$

Then, from (8-22) and (8-25),

$$\tan \theta_M = \frac{1}{2} \left[\sqrt{\frac{\omega_{wp}}{\omega_{w0}}} - \sqrt{\frac{\omega_{w0}}{\omega_{wp}}} \right] \quad (8-26)$$

From this equation, θ_M is seen to be a function only of the ratio ω_{wp}/ω_{w0} . Figure 8-11 is a plot of (8-26). Note also that

$$|D(j\omega_{wm})| = a_0 \frac{\sqrt{1 + (\omega_{wm}/\omega_{w0})^2}}{\sqrt{1 + (\omega_{wm}/\omega_{wp})^2}} \Big|_{\omega_{wm}} = a_0 \sqrt{\frac{1 + \omega_{wp}/\omega_{w0}}{1 + \omega_{w0}/\omega_{wp}}} = a_0 \sqrt{\frac{\omega_{wp}}{\omega_{w0}}} \quad (8-27)$$

It is seen in Figure 8-10 that phase-lead compensation introduces phase lead, which is a stabilizing effect, but also increases the high-frequency gain relative to the low-frequency gain, which is a destabilizing effect. Design using phase-lead compensation is illustrated in Figure 8-12. The phase lead is introduced in the vicinity of the

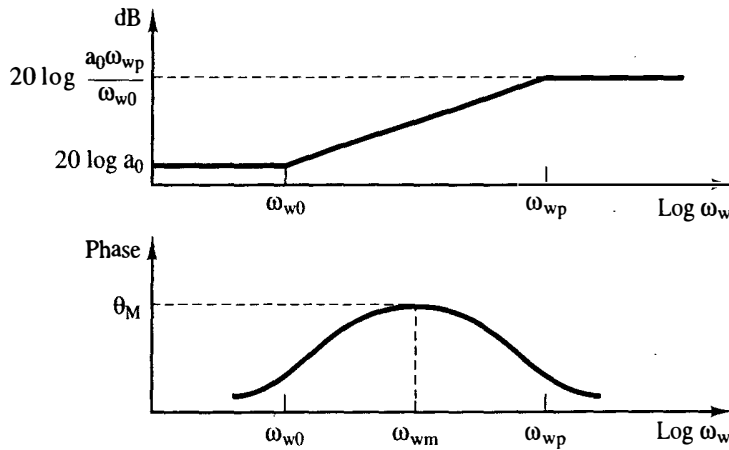


Figure 8-10 Phase-lead digital filter frequency-response characteristics.

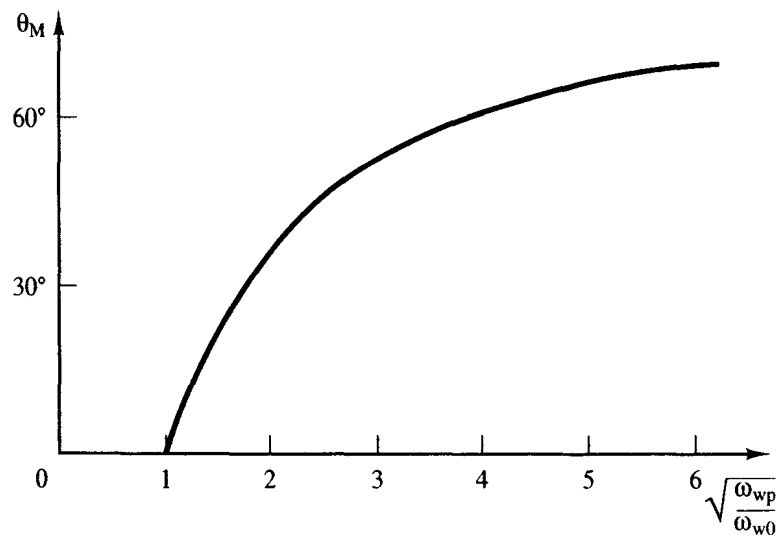


Figure 8-11 Maximum phase shift for a phase-lead filter.

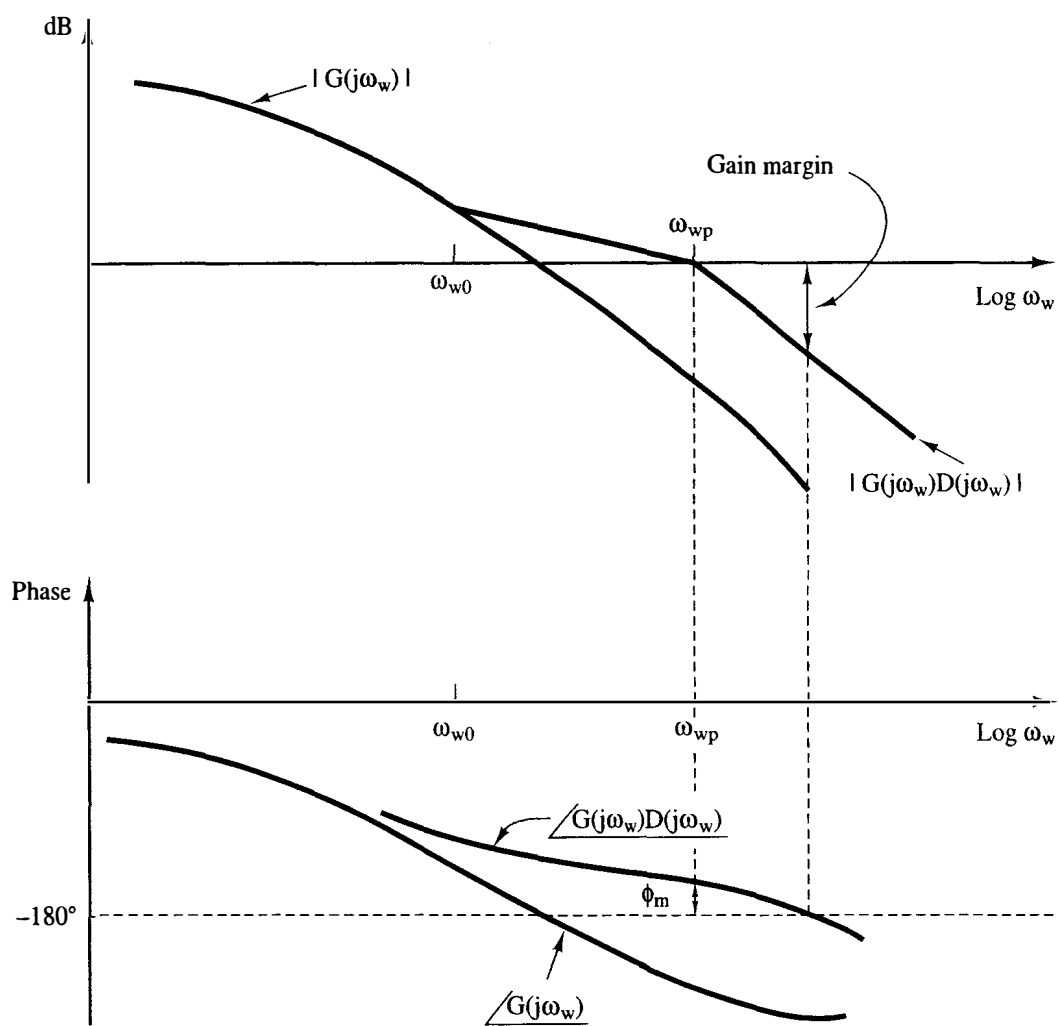


Figure 8-12 Design using phase-lead compensation.

plant's 180° crossover frequency, in order to increase the system's stability margins. Note that system bandwidth is also increased, resulting in a faster time response. For Figure 8-12, the compensator dc gain is unity.

The design of phase-lead compensation tends to be a trial-and-error procedure since, in the frequency range that the stabilizing phase lead is added, destabilizing gain is also added. Given in the next section is a procedure that will yield a specified phase margin, but has no control on the gain margin.

8.6 PHASE-LEAD DESIGN PROCEDURE

Presented in this section is a design procedure that will yield a specified phase margin in a discrete control system, provided that the designed system is stable. That is, the procedure will set the gain and phase of the open-loop function to specified values at a given frequency, and we choose the specified gain to be 0 dB and the specified phase to be $(180^\circ + \phi_m)$, where ϕ_m is the desired phase margin. Thus the procedure does not determine the gain margin, and may in fact result in an unstable system. Then, as a later step in the design procedure, it will be necessary to check the gain margin to ensure that it is adequate.

The characteristic equation for the system of Figure 8-6 is

$$1 + D(w)G(w) = 0$$

where $D(w)$ is given by (8-13). Our design problem may then be stated with respect to this system:

Determine $D(w)$ in (8-28) such that, at some frequency ω_{w1} ,

$$D(j\omega_{w1})G(j\omega_{w1}) = 1 \angle 180^\circ + \phi_m \quad (8-28)$$

and, in addition, the system possesses an adequate gain margin.

The design equations will now be developed. Let $D(w)$ be expressed as

$$D(w) = \frac{a_1 w + a_0}{b_1 w + 1} = a_0 \left[\frac{1 + w/(a_0/a_1)}{1 + w/(b_1)^{-1}} \right] \quad (8-29)$$

where a_0 is the compensator dc gain. Then, in (8-13), (8-14), and (8-15),

$$\omega_{w0} = \frac{a_0}{a_1}, \quad \omega_{wp} = \frac{1}{b_1} \quad (8-30)$$

This notation is more convenient than that used for the phase-lag compensator.

From (8-28) we see that $D(j\omega_{w1})$ must satisfy the relationships

$$|D(j\omega_{w1})| = \frac{1}{|G(j\omega_{w1})|} \quad (8-31)$$

and

$$\angle D(j\omega_{w1}) = 180^\circ + \phi_m - \angle G(j\omega_{w1})$$

where the symbol $\angle D$ denotes the angle associated with the complex number D . Let the angle associated with $D(j\omega_{w1})$ be denoted as θ ; that is,

$$\theta = \angle D(j\omega_{w1}) = 180^\circ + \phi_m - \angle G(j\omega_{w1}) \quad (8-32)$$

In the equation for $D(w)$, (8-29), we see that there are three unknowns, namely, a_0 , a_1 , and b_1 . Equations (8-31) and (8-32) give two constraints on $D(j\omega_{w1})$. Solving these two equations for a_1 and b_1 as functions of a_0 and ω_{w1} yields

$$a_1 = \frac{1 - a_0 |G(j\omega_{w1})| \cos \theta}{\omega_{w1} |G(j\omega_{w1})| \sin \theta} \quad (8-33a)$$

$$b_1 = \frac{\cos \theta - a_0 |G(j\omega_{w1})|}{\omega_{w1} \sin \theta} \quad (8-33b)$$

(See Appendix I.) In the case that the sensor transfer function $H(s)$ is not unity, replace $G(j\omega_{w1})$ with $\overline{GH}(j\omega_{w1})$ in (8-31), (8-32), and (8-33).

If the compensator coefficients satisfy the preceding equations, the Nyquist diagram will pass through the point $1/-180^\circ + \phi_m$. If the designed system is stable, this system has the required phase margin. However, nothing in the design procedure guarantees stability. Thus, once the coefficients are calculated, the Bode diagram (or Nyquist diagram) must be calculated to determine if the closed-loop system is stable.

This design procedure requires that the compensator dc gain a_0 and the system phase-margin frequency ω_{w1} be chosen. Then (8-33) determines the compensator coefficients a_1 and b_1 . The compensator dc gain is usually determined by steady-state specifications for the control system. The frequency ω_{w1} can be approximately determined in the following manner. Since the compensator is phase lead, the angle θ of (8-32) must be positive. Thus, from (8-32), $\theta > 0$ yields

$$1. \angle G(j\omega_{w1}) < 180^\circ + \phi_m$$

Also, $|D(j\omega_{w1})| > a_0$ from Figure 8-10, and, from (8-31),

$$2. |G(j\omega_{w1})| < 1/a_0$$

In addition, the coefficient b_1 must be positive, to ensure a stable controller. Thus, from (8-33),

$$3. \cos \theta > a_0 |G(j\omega_{w1})|$$

Hence, the phase-margin frequency ω_{w1} must be chosen to satisfy these three constraints.

Problem 8-9).

Example 8.2



Consider again the system of Example 8.1, whose frequency response is given in Figure 8-8 and Table 8-1. A phase margin of 55° is to be achieved, and a unity-dc-gain phase-lead compensator will be employed. Consider Table 8-1. We must choose a frequency ω_{w1} such that $\angle G(j\omega_{w1}) < -125^\circ$ (constraint 1) and $|G(j\omega_{w1})| < 1$ (constraint 2). We rather arbitrarily choose $\omega_{w1} = 1.200$ to satisfy these two constraints, and from Table 8-1, $G(j\omega_{w1}) = 0.4576/-172.9^\circ$. From (8-32),

$$\theta = 180^\circ + 55^\circ - (-172.9^\circ) = 407.9^\circ = 47.9^\circ$$

Constraint 3 yields $\cos 47.9^\circ = 0.670 > 0.4576$, and all constraints are satisfied. Hence, from (8-33),

$$a_1 = \frac{1 - (1)(0.4576) \cos(47.9^\circ)}{(1.2)(0.4576) \sin(47.9^\circ)} = 1.701$$

$$b_1 = \frac{\cos(47.9^\circ) - (1)(0.4576)}{(1.2) \sin(47.9^\circ)} = 0.2387$$

and thus, from (8-29),

$$D(w) = \frac{1 + 1.701w}{1 + 0.2387w} = \frac{1 + w/0.5879}{1 + w/4.187}$$

We then obtain the filter transfer function from (8-14):

$$D(z) = \frac{6.539(z - 0.9710)}{z - 0.8106}$$

Calculation of the system open-loop frequency response shows that this compensator results in a phase margin of 55° and a gain margin of 12.3 dB. If we choose ω_{w1} as a value different from 1.2 above, the phase margin will remain at 55° , but the gain margin will be different. If ω_{w1} is chosen larger, then θ , the angle of $D(j\omega_{w1})$, is larger (see Table 8-1), and the ratio ω_{wp}/ω_{w0} is larger. Thus, from Figure 8-10, the high-frequency gain increases, which increases the system bandwidth. Choosing ω_{w1} less than 1.2 will reduce the system bandwidth. The calculations in this example are implemented in the MATLAB program:

```

disp(' Enter the required parameters: ')
a0=input('
phim=input('
ww1=input('
gww1mag=input('
phasegww1=input('
thetad=180+phim-phasegww1;
thetar=thetad*pi/180;
a0
a1=(1-a0*gww1mag*cos(thetar))/(ww1*gww1mag*sin(thetar))
disp(' ')

b1=(cos(thetar)-a0*gww1mag)/(ww1*sin(thetar))

Filter dc gain    a0 = ');
Desired phase margin    phim = ');
Phase margin frequency    ww1 = ');
Magnitude    G(jww1) = ');
Phase    G(jww1) = ');

```

The coefficient quantization problem observed for the phase-lag filter generally does not occur in phase-lead filters. For a phase-lag filter, the pole and zero are almost coincident, making their placement critical. For the phase-lead filter, the pole and zero are well separated, and any small shift in either one has little effect on the filter frequency response.

Examples 8.1 and 8.2 illustrate simple phase-lag and phase-lead compensation. The effect of phase-lag compensation is to reduce the open-loop gain at higher frequencies, which in turn reduces system bandwidth. The open-loop gain at lower frequencies is not reduced (can be increased), and thus steady-state errors are not increased (can be reduced). The effect of phase-lead compensation is to increase open-loop gain at higher frequencies, and thus increases system bandwidth.

The open-loop frequency responses of the system of the foregoing two examples without compensation, with phase-lag compensation, and with phase-lead compensation are shown in Figure 8-13. Figure 8-14a gives the magnitudes of the closed-loop frequency responses for the two examples and for the uncompensated system. Note the reduced bandwidth for phase-lag compensation and increased bandwidth for phase-lead compensation. Shown in Figure 8-14b are the step responses of the closed-loop system with phase-lag compensation and with phase-lead compensation, obtained by digital simulation. Note that the system step response is much faster for the phase-lead case, because of the increased system bandwidth. Table 8-2 gives several step-response characteristics (see Figure 8-1) for the two cases. Note that while the phase margins are equal, the peak overshoots are not. Thus phase margin alone does not determine peak overshoot.

Open-loop frequency response should not be confused with *closed-loop* frequency response. Figure 8-13 is a plot of the systems open-loop frequency responses, and Figure 8-14a gives the closed-loop frequency responses for the same systems. Table 8-2 gives the closed-loop bandwidths, which were obtained from Figure 8-14a. These bandwidths are not available from Figure 8-13 without extensive calculations. We design using the open-loop frequency response, for convenience, but the closed-loop frequency response more clearly indicates the input-output characteristics of the system.

An additional point should be made concerning phase-lead compensation. From Figure 8-10 we see that the high-frequency gain of the digital filter can be quite large. Hence if the control system is burdened with high-frequency noise, the phase-lead compensation may lead to noise problems. If this is the case, some compromise in design may be required. One solution involves the use of a phase-lag compensator cascaded with a phase-lead compensator. The lag compensation is employed to realize a part of the required stability margins, thus reducing the amount of phase-lead compensation required. This compensation is discussed in the next section.

A second possible solution to this noise problem would be to add one or more poles to the filter transfer function. The pole (or poles) are placed at high frequencies such that the phase lag introduced by the poles does not significantly decrease the

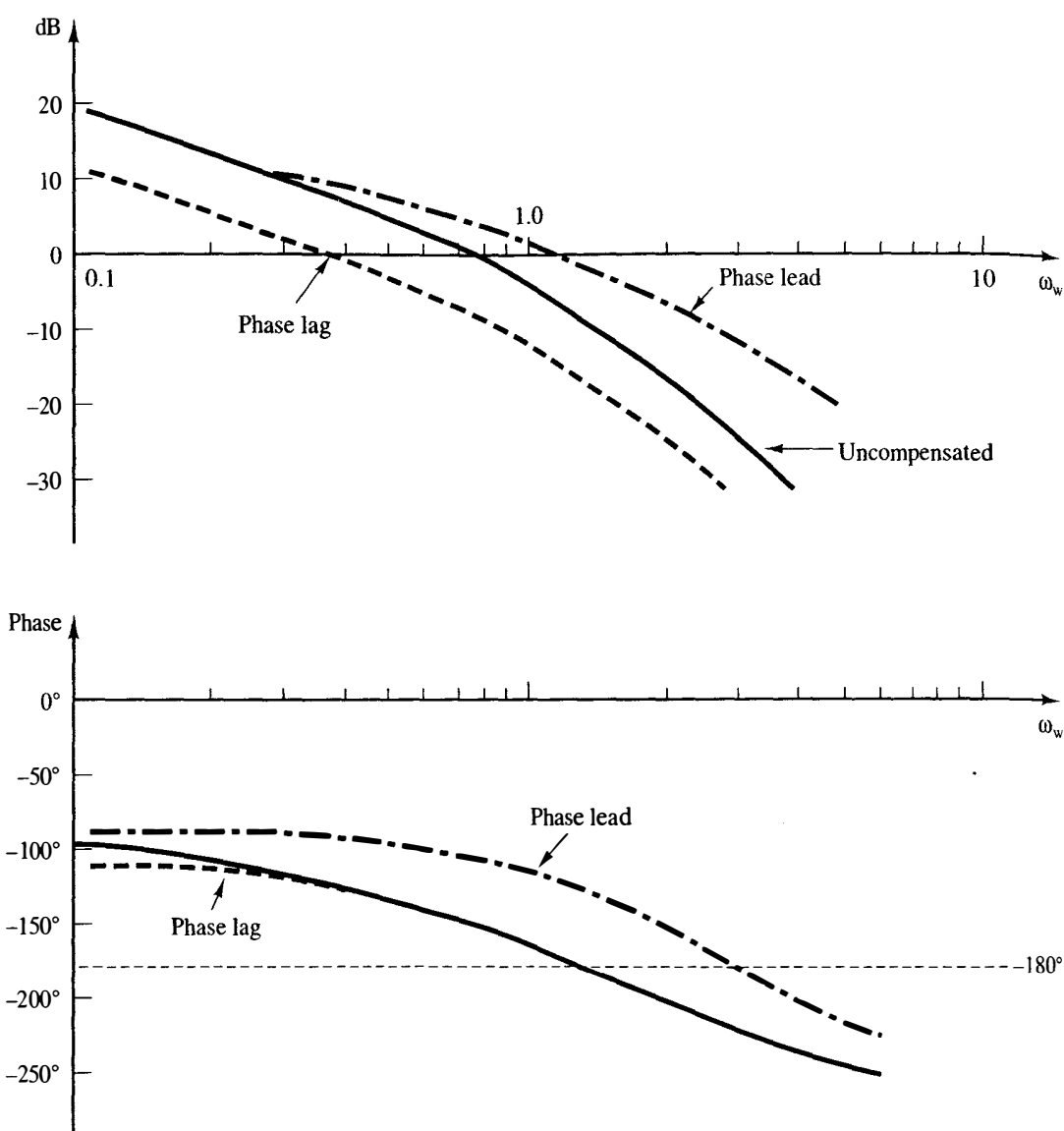


Figure 8-13 Open-loop frequency responses for systems of Examples 8.1 and 8.2.

system stability margins. The required transfer function for a single pole is of the form

$$D_h(w) = \frac{1}{1 + w/\omega_{wp1}}$$

and the total filter transfer function is then

$$D(w) = \frac{a_1 w + a_0}{b_1 w + 1} \left[\frac{1}{1 + w/\omega_{wp1}} \right]$$

The resultant filter frequency response is as shown in Figure 8-15, for $a_0 = 1$.

Another problem originates in the large high-frequency gain of the phase-lead

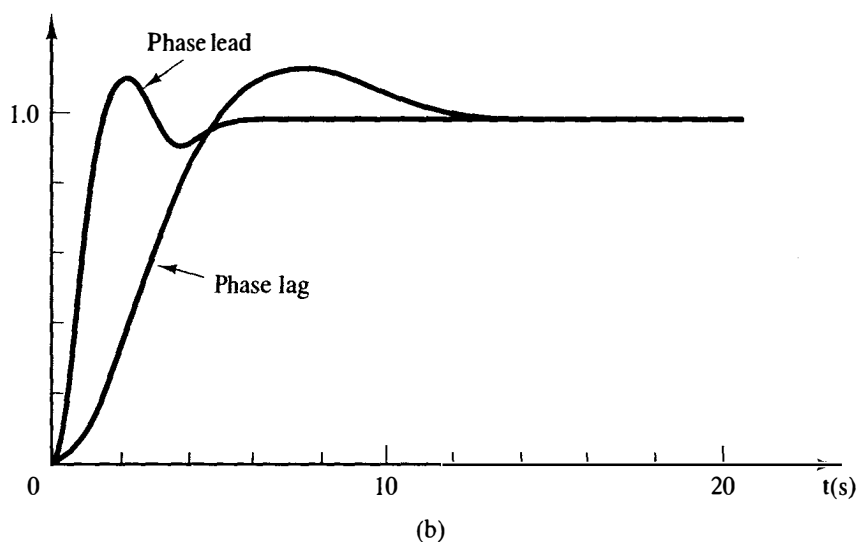
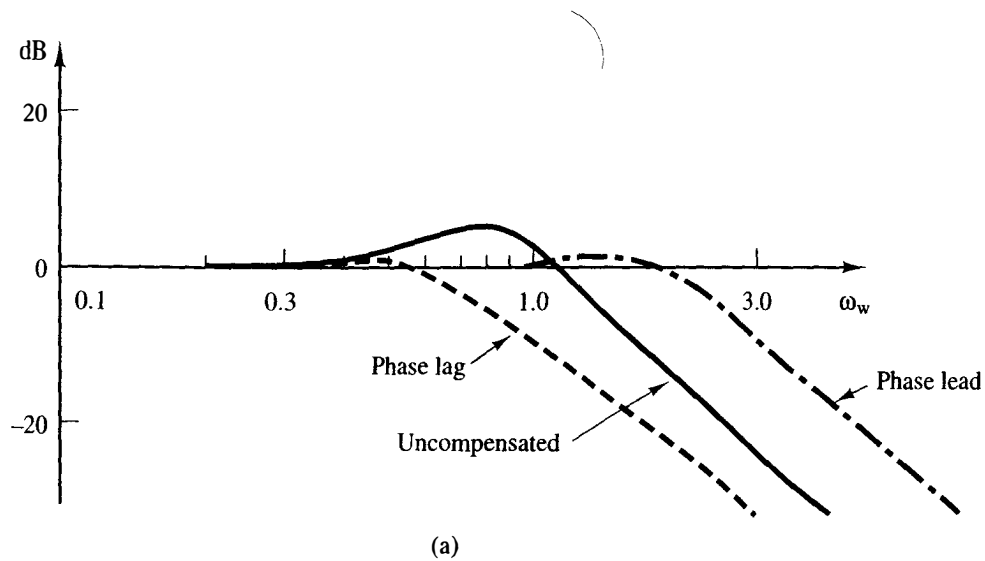


Figure 8-14 (a) Closed-loop frequency responses; (b) unit-step responses for Examples 8.1 and 8.2.

TABLE 8-2 STEP-RESPONSE CHARACTERISTICS

	Phase lag	Phase lead
Steady-state error	0	0
Percent overshoot	15	13
Rise time, t_r (s)	3.2	1.0
t_p (s)	7.3	2.2
t_s , for $d = 0.05$ (s)	11.7	4.7
Bandwidth (rad/s)	0.66	2.20

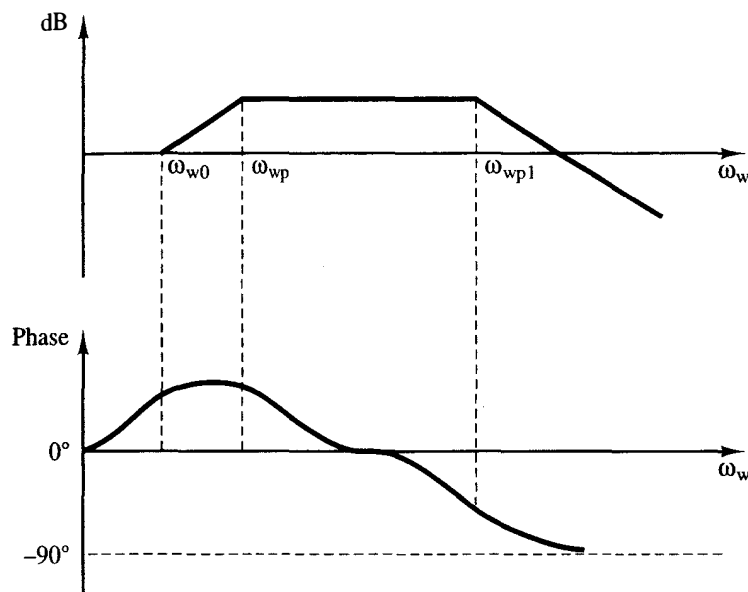


Figure 8-15 Phase-lead filter with an added pole.

compensator. This problem is evident in the filter transfer function of Example 8.2. For this example,

$$D(z) = \frac{6.539(z - 0.9710)}{z - 0.8106} = 6.539 - 1.048z^{-1} + \dots$$

Thus a step change of 1 unit in the filter input results in a step change of 6.539 units in the filter output. This large signal out of the filter may force the plant into a nonlinear region of operation (e.g., an amplifier may saturate), or the digital-to-analog converter may saturate. Since the design is based on a linear plant model, the effects of forcing the system into nonlinear regions of operation will not, in general, be obvious.

In summary, some possible advantages of phase-lag compensation are:

1. The low-frequency characteristics are maintained or improved.
2. The stability margins are improved.
3. The bandwidth is reduced, which is an advantage if high-frequency noise is a problem. Also, for other reasons, reduced bandwidth may be an advantage.

Some possible disadvantages are:

1. The reduced bandwidth may be a problem in some systems.
2. The system transient response will have one very slow term. This will become evident when root-locus design is covered in Section 8.11.
3. Numerical problems with filter coefficients may result.

For phase-lead compensation, some possible advantages are:

1. Stability margins are improved.
2. High-frequency performance, such as speed of response, is improved.
3. Phase-lead compensation is *required* to stabilize certain types of systems.

Some possible disadvantages are:

1. Any high-frequency noise problems are accentuated.
2. Large signals may be generated, which may damage the system or at least result in nonlinear operation of the system. Since the design assumed linearity, the results of the nonlinear operation will not be immediately evident.

8.7 LAG-LEAD COMPENSATION

In the preceding sections, only simple first-order compensators were considered. In many system design projects, however, the system specifications cannot be satisfied by a first-order compensator. In these cases higher-order filters must be used. To illustrate this point, suppose that smaller steady-state errors to ramp inputs are required for the system of Example 8.2. Then the low-frequency gain of the system must be increased. If phase-lead compensation is employed, this increase in gain must be reflected at all frequencies (see Figure 8-10). It is then unlikely that one first-order section of phase-lead compensation can be designed to give adequate stability margins. One solution to this problem would be to cascade two first-order phase-lead filters. However, if noise in the control system is a problem, the increased gain at high frequencies may lead to noise problems. A different approach is to cascade a phase-lag filter with a phase-lead filter. This filter is usually referred to as a lag-lead compensator.

A lag-lead filter has the characteristics shown in Figure 8-16. The purpose of

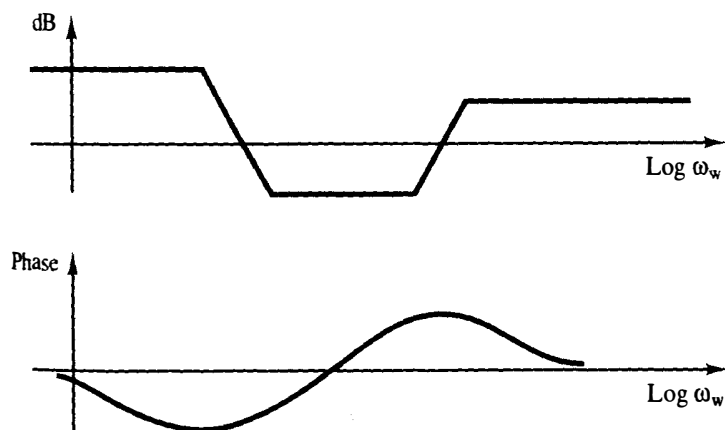


Figure 8-16 Frequency response of a lag-lead filter.

the lag section is to increase the low-frequency gain, and the lead section increases the bandwidth and the stability margins. The lag-lead filter will now be illustrated by an example.

Example 8.3

In this example, the system of Examples 8.1 and 8.2 will again be considered. The steady-state error to a unit ramp input is given by

$$[eq. (6-21)] \quad e_{ss}(kT) = \frac{T}{\lim_{z \rightarrow 1} (z-1)G(z)}$$

From Example 8.1,

$$\lim_{z \rightarrow 1} (z-1)G(z) = \lim_{z \rightarrow 1} \frac{(z-1)^2}{z} \left[\frac{0.05z}{(z-1)^2} - \frac{1.5z}{z-1} + \frac{2z}{z-0.9512} - \frac{0.5z}{z-0.9048} \right] = 0.05$$

Hence, from (6-21),

$$e_{ss}(kT) = \frac{0.05}{0.05} = 1$$

Suppose that the design specifications require a steady-state error to a unit ramp input of 0.50 and a phase margin of 55° . We will use a phase-lag filter, $D_1(z)$, to increase the low-frequency gain by a factor of 2, to satisfy the steady-state error criterion. Then we design a phase-lead filter, $D_2(z)$, to yield the 55° phase margin. We will choose the pole-zero locations of $D_1(z)$ to be the same as those of the phase-lag filter in Example 8.1. Then

$$\lim_{z \rightarrow 1} D_1(z) = \lim_{z \rightarrow 1} \frac{K_d(z-0.998202)}{z-0.999300} = 2$$

From this expression we see that $K_d = 0.7786$, or

$$D_1(z) = \frac{0.7786(z-0.998202)}{z-0.999300}$$

To design the phase-lead filter, we must calculate the frequency response $D_1(z)G(z)$. The equations of Section 8.6 can then be utilized to find the phase-lead filter transfer function $D_2(z)$. We use $\omega_{w1} = 1.20$, as in Example 8.2. Calculation of the required frequency response yields

$$D_1(w)G(w) \Big|_{w=j1.20} = 0.365 / -173.9^\circ$$

We now substitute these values into (8-32) and (8-33), with $G(j\omega_1)$ in these equations replaced with $D(j\omega_1)G(j\omega_1)$ above. Hence, from (8-32),

$$\theta = 180^\circ + 55^\circ - (-173.9^\circ) = 408.9^\circ = 48.9^\circ$$

From (8-29) and (8-33), with $a_0 = 1$,

$$a_1 = \frac{1}{\omega_{w0}} = \frac{1 - (1)(0.365) \cos(48.9^\circ)}{(1.2)(0.365) \sin(48.9^\circ)} = 2.303 = \frac{1}{0.434}$$

and from (8-33b),

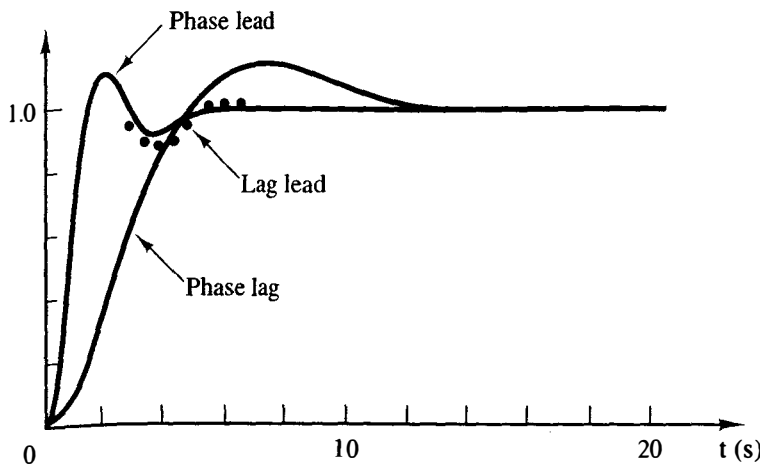


Figure 8-17 Step responses for Examples 8.1, 8.2, and 8.3.

$$b_1 = \frac{1}{\omega_{wp}} = \frac{\cos(48.9^\circ) - (1)(0.365)}{(1.2) \sin(48.9^\circ)} = 0.3233 = \frac{1}{3.093}$$

Then $D_2(w) = (1 + w/0.434)/(1 + w/3.093)$. Thus, from (8-14),

$$D_2(z) = \frac{6.68(z - 0.9785)}{z - 0.857}$$

Calculation of the compensated system open-loop frequency response shows that the compensated system has a phase margin of 55° and a gain margin of 11.2 dB. The step response for this system is plotted in Figure 8-17, together with those from Examples 8.1 and 8.2. Note that the step responses of the phase-lead system and the lag-lead system are approximately the same; however, the steady-state error for a ramp input for the lag-lead system is only one-half that for the phase-lead system. The total filter transfer function is

$$D(z) = D_1(z)D_2(z) = \frac{5.20(z - 0.998202)(z - 0.9785)}{(z - 0.999300)(z - 0.857)}$$

A sketch of the Bode diagram for this filter is shown in Figure 8-18.

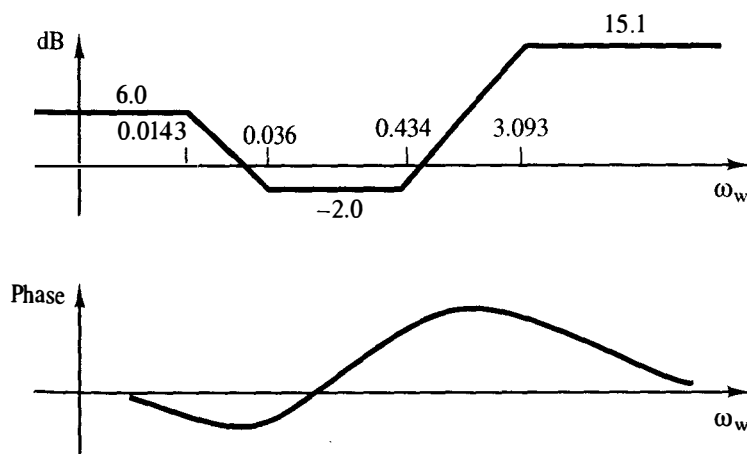


Figure 8-18 Frequency response for the lag-lead filter for Example 8.3.

8.8 INTEGRATION AND DIFFERENTIATION FILTERS

A somewhat different controller is presented in this and the following section. This controller is used extensively in the chemical processing industry and to a lesser degree in the aerospace industry.

To introduce this controller, we first consider a technique for digital-filter integration. Suppose that we desire to integrate a signal $e(t)$ digitally, and to accomplish this, we will use the trapezoidal technique [6]. The trapezoidal rule is illustrated in Figure 8-19. Let $m(kT)$ be the numerical integral of $e(t)$. Then, from Figure 8-19, the value of the integral at $t = (k + 1)T$ is equal to the value at kT plus the area added from kT to $(k + 1)T$. From Figure 8-19,

$$m[(k + 1)T] = m(kT) + \frac{T}{2}\{e[(k + 1)T] + e(kT)\} \quad (8-34)$$

Taking the z -transform, we obtain

$$zM(z) = M(z) + \frac{T}{2}[zE(z) + E(z)] \quad (8-35)$$

Thus

$$\frac{M(z)}{E(z)} = \frac{T}{2} \left[\frac{z + 1}{z - 1} \right] \quad (8-36)$$

Hence (8-36) is the transfer function of a discrete integrator. Of course, there are many other discrete transfer functions that may be used to integrate a number sequence (see Chapter 11).

Now consider a technique for the digital-filter differentiation of a function $e(t)$. Figure 8-20 illustrates one method. The slope of $e(t)$ at $t = kT$ is approximated to be the slope of the straight line connecting $e[(k - 1)T]$ with $e(kT)$. Then, from Figure 8-20, letting the numerical derivative of $e(t)$ at $t = kT$ be $m(kT)$, we can write

$$m(kT) = \frac{e(kT) - e[(k - 1)T]}{T} \quad (8-37)$$

The z -transform of this equation yields

$$\frac{M(z)}{E(z)} = \frac{z - 1}{Tz} \quad (8-38)$$

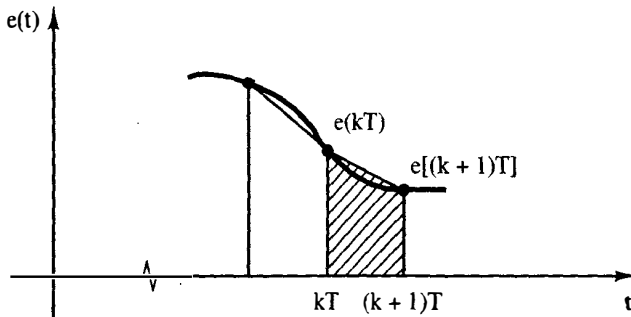


Figure 8-19 Trapezoidal rule for numerical integration.

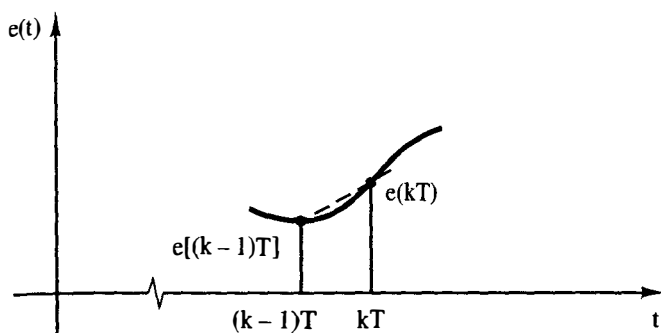


Figure 8-20 Illustration of numerical differentiation.

Note that this differentiator is the reciprocal of the transfer function for the rectangular-rule integrator (see Problem 2-15). In general, the reciprocal of the transfer function of a numerical integrator can be used as the transfer function of a numerical differentiator. Numerical differentiation and integration are covered in detail in Chapter 11.

It is evident from the discussion above that many different discrete transfer functions may be used for numerical integration or numerical differentiation. For the development in this section, we use (8-36) and the reciprocal of this transfer function for integration and differentiation, respectively. Then, from (8-36), letting $D_I(z)$ be the integrator transfer function, we obtain

$$D_I(w) = \frac{T}{2} \left[\frac{z+1}{z-1} \right]_{z=[1+(T/2)w]/[1-(T/2)w]} = \frac{1}{w} \quad (8-39)$$

Letting $D_D(z)$, the differentiator transfer function, be the reciprocal of (8-39),

$$D_D(w) = D_D(z) \Big|_{z=[1+(T/2)w]/[1-(T/2)w]} = w \quad (8-40)$$

Recall that for continuous systems, a differentiator has a transfer function of s , and an integrator a transfer function $1/s$. The frequency responses are obtained by replacing s with $j\omega$.

The frequency responses of the discrete transfer functions (8-39) and (8-40) will now be compared to those of the continuous integrator and differentiator. From (7-10),

$$\omega_w = \frac{2}{T} \tan \frac{\omega T}{2} \quad (8-41)$$

For $\omega T/2$ small, we see that

$$\omega_w \approx \omega \quad (8-42)$$

For the integrator transfer function (8-39),

$$D_I(j\omega_w) = \frac{1}{j\omega_w} \quad (8-43)$$

Substituting (8-42) into (8-43), we obtain

$$D_I(j\omega) \approx \frac{1}{j\omega} \quad (8-44)$$

In a similar manner,

$$D_D(j\omega_w) = j\omega_w \approx j\omega$$

These approximations are good provided that since $\omega_s = 2\pi/T$,

$$\tan\left(\omega \frac{T}{2}\right) = \tan\left(\pi \frac{\omega}{\omega_s}\right) \approx \pi \frac{\omega}{\omega_s}$$

If this expression is satisfied, we would expect to obtain accurate differentiation and integration for (8-40) and (8-39), respectively.

8.9 PID CONTROLLERS

We will now discuss a frequency-response design technique that considers phase-lead phase-lag controllers from a somewhat different viewpoint. The resultant controller, called a PID (proportional-plus-integral-plus-derivative) controller, has the block diagram shown in Figure 8-21. This controller is a special type of the lag-lead controller.

The transfer function of a digital PID controller, using the integrator and differentiator transfer functions developed in the preceding section, is given by (see Figure 8-21)

$$D(w) = K_p + \frac{K_I}{w} + K_D w \quad (8-45)$$

In this expression, K_p is the gain in the proportional path, K_I the gain in the integral path, and K_D the gain in the derivative path.

Consider first a proportional-plus-integral (PI) controller. The filter transfer function is

$$D(w) = K_p + \frac{K_I}{w} = \frac{K_p w + K_I}{w} = K_I \frac{1 + w/\omega_{w0}}{w} \quad (8-46)$$

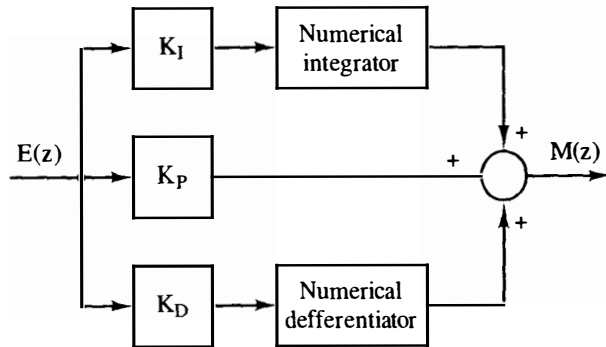


Figure 8-21 Discrete PID controller.

where $\omega_{w0} = K_I/K_p$. Note that this is a phase-lag filter of the type given in (8-13), with the pole placed at $\omega_{wp} = 0$. The filter frequency response is illustrated in Figure 8-22. Note that the PI controller low-frequency gain is increased (infinite at $\omega_w = 0$), and thus steady-state errors are reduced. The purpose is then the same as that of the phase-lag controller of (8-13): to increase stability margins and/or reduce steady-state errors.

Consider next the PD (proportional-plus-derivative) controller. The filter transfer function is

$$D(w) = K_p + K_D w = K_p \left(1 + \frac{w}{\omega_{w0}} \right) \quad (8-47)$$

where $\omega_{w0} = K_p/K_D$. Note that this is a phase-lead controller of the type given in (8-13), with the pole placed at $\omega_{wp} = \infty$. The filter frequency response is illustrated in Figure 8-23. The purposes of the PD controller are to add positive phase angles to the open-loop frequency response so as to improve system stability, and to increase closed-loop system bandwidth so as to increase the speed of response. The effects of the PD controller appear at high frequencies, as opposed to the low-frequency effects of the PI controller.

The PID filter is a composite of the two filters discussed above and has the transfer function

$$D(w) = K_p + \frac{K_I}{w} + K_D w \quad (8-48)$$

The frequency response of the PID controller is illustrated in Figure 8-24. From (8-48),

$$D(w) = \frac{K_D w^2 + K_p w + K_I}{w} = \frac{K_I(1 + w/\omega_{w0})(1 + w/\omega_{w02})}{w}$$

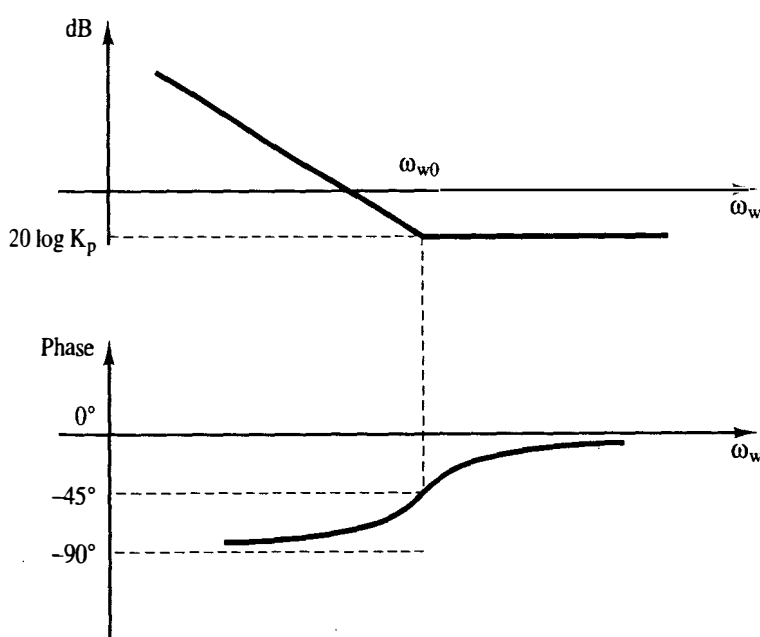
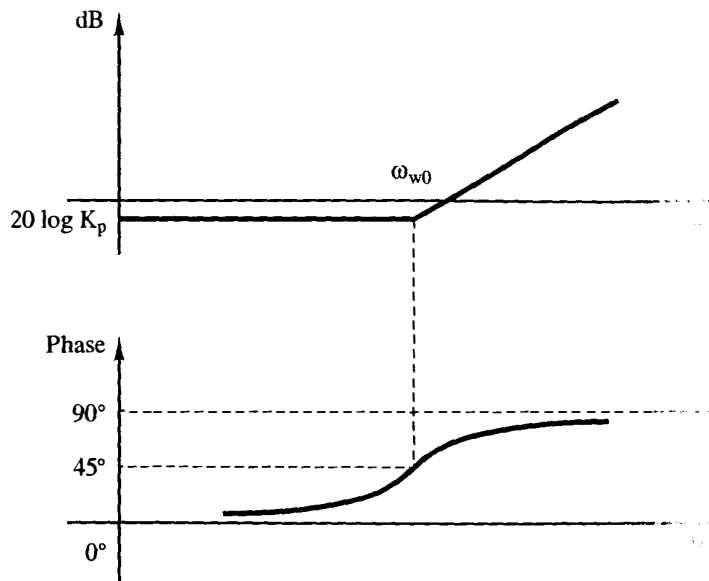


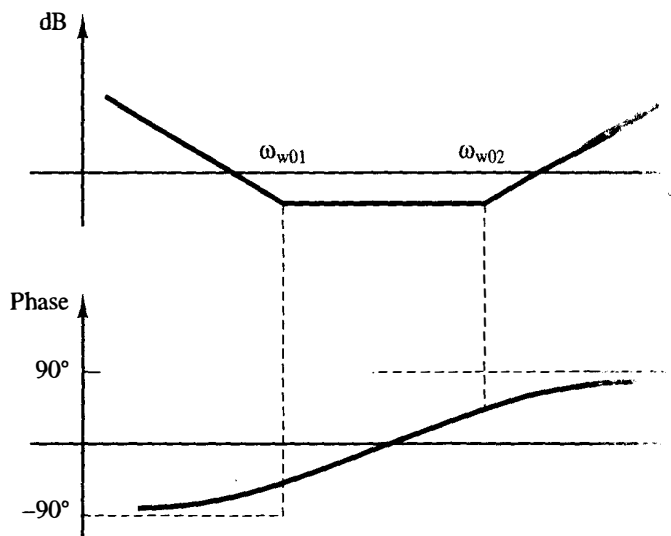
Figure 8-22 Frequency response for a PI controller.



8-23 Frequency response for a PI controller.

Hence the filter has two zeros, ω_{w01} and ω_{w02} , and 2 poles, with one at $\omega_w = 0$ and the other at $\omega_w = \infty$. The design techniques can follow those given for phase-lag (the PI part) and phase-lead (the PD part) filters. However, a different design procedure will be developed in the following section. Since the K term is common to both the PI and PD parts, the design of the PD part does not affect the PI part, and vice versa.

A problem is obvious in the use of the PID filter. As is seen in Figure 8-24, the filter gain continues to increase without limit as frequency increases. As ω_w approaches infinity, ω approaches $\omega_s/2$ and z approaches -1 . In general, $G(z)$ does not have a zero at $z = -1$. Thus $|G(z)D(z)|$ will approach infinity as z approaches -1 , which will probably lead to encirclements of the -1 point on the system Nyquist diagram. Hence the system will be unstable. & the compensator of (8-48) is not practical. To avert this problem, a pole (or poles) must be added to the derivative term, resulting in the filter transfer function:



8-24 Frequency response for a PID controller.

$$D(w) = K_p + \frac{K_I}{w} + \frac{K_D w}{\prod(1 + w/\omega_{wpi})} \quad (8-49)$$

However, the usual solution to this problem is to add a single pole using the differentiator of Figure 8-20, which has the transfer function given in (8-38).

$$D_D(w) = \frac{z-1}{Tz} \Big|_{z=[1+(T/2)w]/[1-(T/2)w]} = \frac{w}{1+(T/2)w} = \frac{w}{1+(w/\omega_{wp})} \quad (8-50)$$

Hence this algorithm for differentiation adds a pole at $2/T$; and since $\omega_s = 2\pi/T$, then

$$\omega_{wp} = -\frac{2}{T} = -\frac{\omega_s}{\pi} \quad (8-51)$$

This pole position is normally well outside the bandwidth of the system and will contribute very little phase lag at the frequency that the phase margin occurs. Hence in many cases this pole can be ignored (at least initially) in the design process. If high-frequency noise is a problem in the control system, additional poles may be required in the derivative term.

A practical PID filter transfer function as a function of z is, from (8-45), (8-36), and (8-38),

$$D(z) = K_p + K_I \frac{T}{2} \left[\frac{z+1}{z-1} \right] + K_D \left[\frac{z-1}{Tz} \right] \quad (8-52)$$

This filter uses trapezoidal integration and the differentiation of (8-38) and Figure 8-20. The filter difference equations can be written directly from this equation (see Problem 8-24). In some cases it is possible to determine K_p , K_I , and K_D experimentally using the physical control system, when only a rudimentary knowledge of the plant characteristics is available. An educated guess of K_p , K_I , and K_D is made, which must result in a stable closed-loop system. Then K_p , K_I , and K_D are varied in some systematic manner until an acceptable response is obtained. A somewhat different form of the PID controller transfer function is considered in Problem 8-25.

The PID controller offers an advantage if, for some reason, the sample period T must be changed after the system design has been completed. Suppose that for a given design, T is chosen such that accurate differentiation and integration occur. If for a different value of T , accurate differentiation and integration still occur, the gains K_p , K_I , and K_D will not change. Hence a new design is not required. A design procedure for PID controllers will be developed in the following section.

In this section a design process is developed for PID controllers [6]. The development will closely parallel that of Section 8.6 for phase-lead filters (controllers). We assume initially that the PID controller has a transfer function given by

[eq. (8-45)]
$$D(w) = K_p + \frac{K_I}{w} + K_D w$$

Hence we are ignoring the effects of the required pole of (8-50) in the derivative path; these effects will be considered later. The controller frequency response is given by

$$D(j\omega_w) = K_p + j\left(K_D \omega_w - \frac{K_I}{\omega_w}\right) = |D(j\omega_w)| \angle \theta \quad (8-53)$$

Here, as in Section 8.6, the design problem is to choose $D(w)$ (i.e., choose K_p , K_I , and K_D) such that

$$D(j\omega_{w1})G(j\omega_{w1}) = 1 \angle 180^\circ + \phi_m \quad (8-54)$$

at a chosen frequency ω_{w1} . Now, from (8-53),

$$K_p + j\left(K_D \omega_{w1} - \frac{K_I}{\omega_{w1}}\right) = |D(j\omega_{w1})|(\cos \theta + j \sin \theta) \quad (8-55)$$

where from (8-53) and (8-54),

$$\theta = 180^\circ + \phi_m - \angle G(j\omega_{w1}) \quad (8-56)$$

Therefore, from (8-54) and (8-55),

$$K_p = |D(j\omega_{w1})| \cos \theta = \frac{\cos \theta}{|G(j\omega_{w1})|} \quad (8-57)$$

$$K_D \omega_{w1} - \frac{K_I}{\omega_{w1}} = \frac{\sin \theta}{|G(j\omega_{w1})|} \quad (8-58)$$

In these design equations, if the transfer function of the sensor $H(s)$ is not unity, $G(j\omega_{w1})$ is replaced with $\overline{GH}(j\omega_{w1})$.

The design equations are then (8-56), (8-57), and (8-58). For a given plant $[G(w)]$, the choice of ω_{w1} and ϕ_m uniquely determines K_p , from (8-57). However, K_D and K_I are not uniquely determined, as is evident in (8-58). This results from (8-55) yielding two equations, but with the three unknowns K_p , K_I , and K_D . In satisfying (8-58), in general increasing K_D will increase the bandwidth, while increasing K_I will decrease steady-state errors, *provided* that the system retains an acceptable gain margin. Note that if (8-57) and (8-58) are satisfied, varying K_D and K_I will change the gain margin, while the phase margin remains constant.

Equations (8-57) and (8-58) also apply to the design of PI and PD controllers, with the appropriate gain (K_D or K_I) set to zero. For this case all gains are uniquely determined.

As shown in Section 8.9, in general a pole is required in the derivative term. A commonly used transfer function was given in (8-50), which results in a PID controller transfer function

$$D(w) = K_p + \frac{K_I}{w} + \frac{K_D w}{1 + (T/2)w} \quad (8-59)$$

Then

$$\begin{aligned} D(j\omega_w) &= K_p - j\frac{K_I}{\omega_w} + \frac{K_D j\omega_w}{1 + j\omega_w T/2} \\ &= \left(K_p + \frac{K_D \omega_w^2 (2/T)}{(2/T)^2 + \omega_w^2} \right) + j \left(\frac{K_D \omega_w (2/T)^2}{(2/T)^2 + \omega_w^2} - \frac{K_I}{\omega_w} \right) \end{aligned} \quad (8-60)$$

Hence (8-57) and (8-58) become

$$K_p + \frac{K_D \omega_{w1}^2 (2/T)}{(2/T)^2 + \omega_{w1}^2} = \frac{\cos \theta}{|G(j\omega_{w1})|} \quad (8-61)$$

and

$$\frac{K_D \omega_{w1} (2/T)^2}{(2/T)^2 + \omega_{w1}^2} - \frac{K_I}{\omega_{w1}} = \frac{\sin \theta}{|G(j\omega_{w1})|} \quad (8-62)$$

Note that if $\omega_{w1} \ll 2/T$, these equations reduce to (8-57) and (8-58); otherwise, no simple procedure has been found for calculating K_p , K_I , and K_D . For the PD controller, (8-61) and (8-62) contain only two unknowns and may be solved directly for

$$K_D = \left[\frac{\sin \theta}{|G(j\omega_{w1})|} \right] \left[\frac{(2/T)^2 + \omega_{w1}^2}{(2/T)^2 \omega_{w1}} \right] \quad (8-63)$$

and

$$K_p = \frac{\cos \theta}{|G(j\omega_{w1})|} - \frac{K_D \omega_{w1}^2 (2/T)}{(2/T)^2 + \omega_{w1}^2} \quad (8-64)$$

Two examples of PID design will now be given.

Example 8.4



First the design problem of Example 8.3 will be repeated, but a PI filter will be utilized. In Example 8.3 we required a 55° phase margin and a steady-state error to a unit ramp input of 0.5. Since $D(z)$ adds a pole at $z = 1$ to the one already present in $G(z)$, then $D(z)G(z)$ has two poles at $z = 1$. Thus the steady-state error to a ramp input is zero (see Section 6.5), satisfying the steady-state error design criteria. From Table 8-1, the frequency response of $G(z)$, we choose $\omega_{w1} = 0.400$. Then

$$G(j\omega_{w1}) = G(j0.4) = 2.28 \angle -123.7^\circ$$

From (8-56),

$$\theta = 180^\circ + 55^\circ - (-123.7^\circ) = 358.7^\circ = -1.3^\circ$$

We see then that the phase angle of the filter is very small at the phase-margin frequency, as desired. Then, from (8-57) and (8-58), respectively,

$$K_p = \frac{\cos(-1.3^\circ)}{2.278} = 0.439$$

$$K_I = \left[\frac{-\sin(-1.3^\circ)}{2.278} \right] 0.4 = 0.00398$$

From (8-46), the zero of the PI controller is placed at

$$\omega_{w0} = \frac{K_I}{K_p} = 0.00907$$

which is quite low. Hence we have reduced the system bandwidth considerably. Even though we get good steady-state error response, a relatively long time will be required to achieve it. In fact, a simulation of the system shows that the error to a unit ramp input is 2.0 after 20 s, and has decreased to only 1.5 at the end of 50 s. If we choose $\omega_{w1} = 0.3$, the PI gains are calculated as $K_p = 0.313$ and $K_I = 0.01556$. Then

$$\omega_{w0} = \frac{K_I}{K_p} = 0.0497$$

and the system bandwidth has been increased. For this PI controller, the error to a ramp input is 1.3 after 20 s, and has decreased to 0.11 at the end of 50 s.

A MATLAB program that performs the calculations in this example is given by

```

phim=input(' Desired phase margin:  phim = ');
ww1=input(' Phase margin frequency:  ww1 = ');
gww1mag=input(' Magnitude G(jww1) = ');
phasegww1=input(' Phase G(jww1) = ');
thetad=180+phim-phasegww1;
thetar=thetad*pi/180;
KP=cos(thetar)/gww1mag

KI=-sin(thetar)*ww1/gww1mag

```

This program is easily expandable to include the differentiation term.

Example 8.5

The design problem of Example 8.4 will be repeated, except that in this example a PID controller will be utilized. The design equations (8-57) and (8-58) will be used (i.e., initially the pole in the derivative term will be ignored). From (8-57),

$$K_p = \frac{\cos \theta}{|G(j\omega_{w1})|}$$

Thus the larger we choose ω_{w1} , the smaller is $|G(j\omega_{w1})|$ (see Table 8-1), and hence the larger is K_p . This increases the open-loop gain, which is desirable for many reasons. We choose $\omega_{w1} = 1.2$. Then

$$G(j1.2) = 0.4576 \angle -172.9^\circ$$

From (8-56),

$$\theta = 180^\circ + 55^\circ - (-172.9^\circ) = 407.9^\circ = 47.9^\circ$$

and from (8-57),

$$K_p = \frac{\cos(47.9^\circ)}{0.4576} = 1.465$$

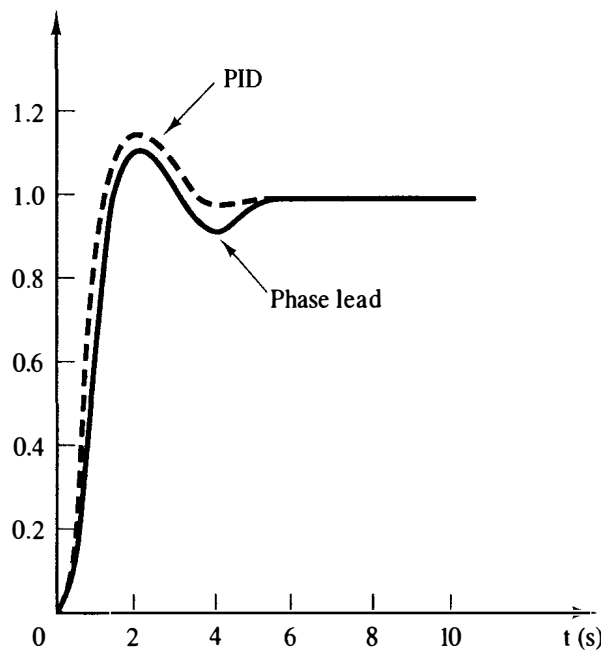


Figure 8-25 Responses of Examples 8.2 and 8.5.

From (8-58),

$$K_D \omega_{w1} - \frac{K_I}{\omega_{w1}} = \frac{\sin \theta}{|G(j\omega_{w1})|}$$

Hence, with $G(j\omega_{w1})$ from above,

$$\begin{aligned} K_D &= \left[\frac{\sin(47.9^\circ)}{0.4576} \right] \frac{1}{1.2} + K_I \left(\frac{1}{1.2} \right)^2 \\ &= 1.351 + 0.694K_I \end{aligned}$$

After some trial and error using a simulation of the system, K_I was chosen as 0.004, and K_D is then 1.354. This choice of gains results in a phase margin of 53.5° and a gain margin of 16 dB, when the pole of (8-50) is added to the derivative path. Thus this pole has little effect on the phase margin. The step response of this system is shown in Figure 8-25, together with the step response of the phase-lead system of Example 8.2. An additional simulation of the system shows that the error to a ramp input has decreased to 0.65 at $t = 20$ s.

8.11 DESIGN BY ROOT LOCUS

In the frequency-response design procedures described above, we attempted to reshape the system open-loop frequency response to achieve certain stability margins, transient-response characteristics, steady-state response characteristics, and so on. Even though design equations were developed, the techniques are still largely trial and error.

A different design technique is presented in this section: the root-locus procedure. Recall that the root locus for a system is a plot of the roots of the system's

characteristic equation as gain is varied. Hence the character of the transient response of a system is evident from the root locus. The design procedure is to add poles and zeros via a digital controller so as to shift the roots of the characteristic equation to more appropriate locations in the z -plane.

Consider again the system of Figure 8-6, which has the characteristic equation

$$1 + KD(z)G(z) = 0 \quad (8-65)$$

where K is the added gain that is to be varied to generate the root locus. Then a point z_a is on the root locus provided that (8-65) is satisfied for $z = z_a$, or that

$$K = \frac{1}{|D(z_a)G(z_a)|} \quad (8-66)$$

$$\angle D(z_a)G(z_a) = \pm 180^\circ \quad (8-67)$$

Since we allow K to vary from zero to infinity, a value of K will always exist such that (8-66) is satisfied. Then the condition for a point z_a to be on the root locus is simply (8-67). If the sensor transfer function $H(s)$ is not unity, $G(z)$ in the foregoing equations is replaced with $\overline{GH}(z)$.

Suppose, as an example, that $D(z) = 1$ and $KG(z)$ is given by

$$KG(z) = \frac{K(z - z_1)}{(z - z_2)(z - z_3)} \quad (8-68)$$

where z_1 , z_2 , and z_3 are all real. Figure 8-26 illustrates the testing of point z to determine if it is on the root locus. If z is on the root locus, then, from (8-67),

$$\theta_1 - \theta_2 - \theta_3 = \pm 180^\circ \quad (8-69)$$

The value of K that places a root of the characteristic equation at this point is, from (8-66),

$$K = \frac{|z - z_2||z - z_3|}{|z - z_1|} \quad (8-70)$$

With this brief discussion of the root locus, we will now consider design procedures. Consider the first-order controller

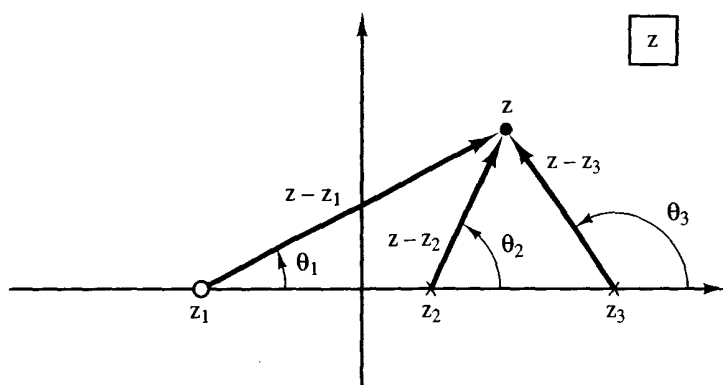


Figure 8-26 Point on the root locus.

$$D(z) = \frac{K_d(z - z_0)}{z - z_p} \quad (8-71)$$

We will require that the filter dc gain equal unity [$D(1) = 1$], so as not to affect the steady-state response; hence

$$K_d = \frac{1 - z_p}{1 - z_0} \quad (8-72)$$

In this design, the gain K is utilized to meet steady-state error requirements. The controller pole is restricted to real values inside the unit circle. For a phase-lead controller $z_0 > z_p$, and thus $K_d > 1$. For a phase-lag controller $z_0 < z_p$ and $K_d < 1$.

We will first consider phase-lag design. We will illustrate the design for the plant, $G(z)$, of (8-68). The uncompensated root locus is sketched in Figure 8-27a. Suppose that the root locations z_a and \bar{z}_a give a satisfactory transient response, but that the loop gain K must be increased to produce smaller steady-state errors, improved disturbance rejection, and so on. We add the controller pole and zero as shown in Figure 8-27b, assuming that $z_3 = 1$. If $z_3 \neq 1$, the controller pole and zero are placed close to $z = 1$. Since the pole and zero are very close to $z = 1$ (recall Example 8.1), the scale in the vicinity of this point is greatly expanded. Hence these two poles and one zero will essentially appear as a single pole as in Figure 8-27a, when determining the root location at z_a . We see, then, that the compensator pole and zero cause the root at z_a to shift only slightly to z'_a , that is, $z'_a \approx z_a$. However,

$$K_c D(z) G(z) = \frac{K_c K_d (z - z_0)(z - z_1)}{(z - z_p)(z - z_2)(z - z_3)}$$

where K_c is the gain in the compensated system. Hence, for a root to appear at z'_a , from (8-66),

$$K_c = \frac{|z'_a - z_p| |z'_a - z_2| |z'_a - z_3|}{K_d |z'_a - z_0| |z'_a - z_1|} \approx \frac{|z_a - z_2| |z_a - z_3|}{K_d |z_a - z_1|} \quad (8-73)$$

Let K_u be the gain required to place the root at z_a in the uncompensated system, as shown in Figure 8-27a. Then

$$K_u = \frac{|z_a - z_2| |z_a - z_3|}{|z_a - z_1|} \quad (8-74)$$

Hence, from (8-73) and (8-74),

$$K_c \approx \frac{K_u}{K_d}$$

Now, K_u is the gain for the uncompensated system, and $K_d < 1$ since the controller is phase lag. Hence the gain for the compensated system, K_c , is greater than that for the uncompensated system, K_u . So we see that phase-lag compensation allows us to increase the open-loop gain while maintaining the approximate same roots in the characteristic equation. Of course, as seen from Figure 8-27b, we have added

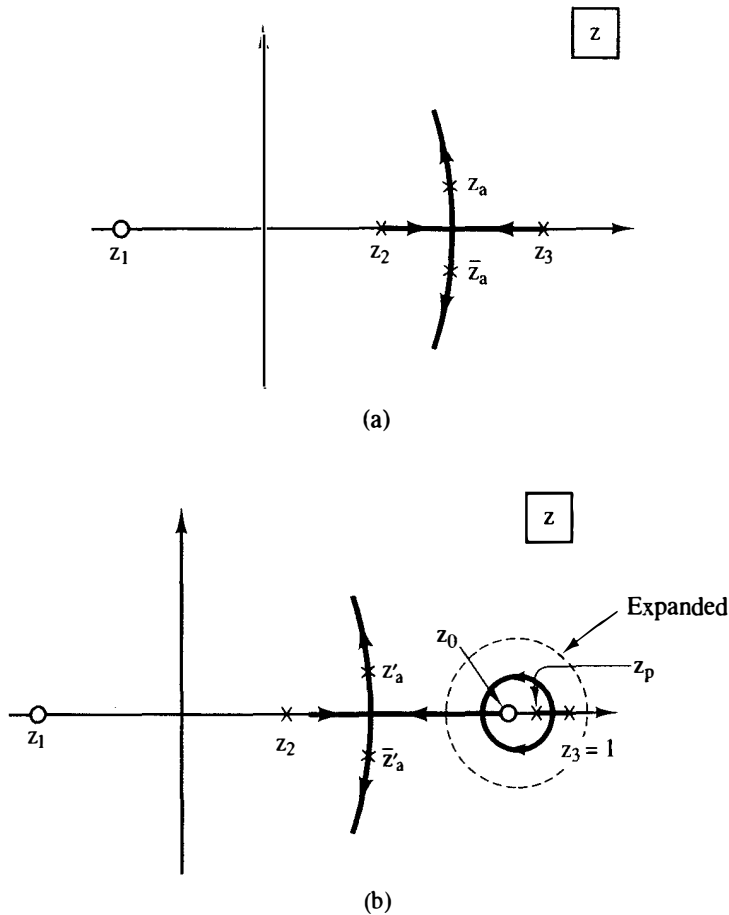


Figure 8-27 Phase-lag design.

a root close to $z = 1$, which has a long time constant. Thus the system will have an increased settling time.

The phase-lag design procedure is then as follows:

1. Choose the desired root location z_a and find K_u , the gain required to place the root at z_a in the uncompensated system.
2. Determine from the design specifications the required compensated-system gain K_c .
3. The compensator gain K_d is calculated from $K_d = K_u/K_c$.
4. Choose the compensator pole z_p sufficiently close to $z = 1$ such that (8-73) is satisfied.
5. From (8-72), the compensator zero is given by

$$z_0 = 1 - \frac{1 - z_p}{K_d} \quad (8-75)$$

From (8-73), this procedure is only approximate.

Example 8.6

Consider the system of Figure 8-28, with $T = 1$ s and

$$G(s) = \frac{1}{s(s+1)} \Rightarrow G(z) = \frac{0.368(z+0.717)}{(z-1)(z-0.368)}$$

The root locus for the uncompensated system is illustrated in Figure 8-27a, with $z_1 = -0.717$, $z_2 = 0.368$, and $z_3 = 1$. This system was shown in Example 7.7 to be critically damped with two poles at $z = 0.65$ for $K = 0.196$. We will design a phase-lag compensator that results in critically damping with two poles at $z \approx 0.65$ with $K = 0.8$. Hence the steady-state errors are reduced by a factor of approximately 4. From the design procedure:

1. $K_u = 0.196$ and $z_a = 0.65$.
2. $K_c = 0.8$.
3. $K_d = \frac{K_u}{K_c} = \frac{0.196}{0.8} = 0.245$
4. Let $z_p = 0.999$.
5. $z_0 = 1 - \frac{1 - 0.999}{0.245} = 0.9959$

Thus

$$D(z) = \frac{0.245(z - 0.9959)}{z - 0.999} = \frac{0.245z - 0.2434}{z - 0.999}$$

A root locus of the compensated system gives two poles at $z \approx 0.65$ for $K = 0.814$. The third pole is at $z = 0.9933$ (see Figure 8-27). A step response yields a rise time of $t_r = 7.5$ s and a 2.6 percent overshoot.

Phase-lead design is illustrated in Figure 8-29. Here, to simplify the discussion, we place the controller zero coincident with the plant pole at $z = z_2$. The controller pole is placed to the left of the zero, which yields the phase-lead controller. Thus the root locus is shifted to the left. The resulting root at z_b has a smaller time constant than that at z_a ; thus the system responds faster (larger bandwidth).

A phase-lead design procedure is as follows:

1. Choose the desired root location z_b .

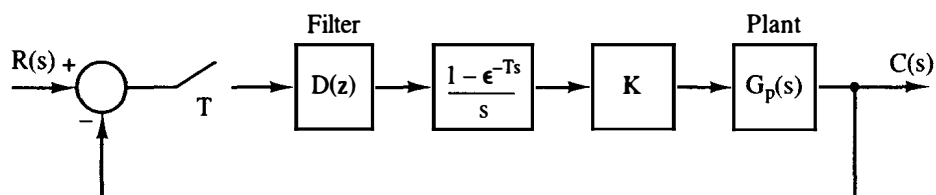


Figure 8-28 System for Examples 8.6 and 8.7.

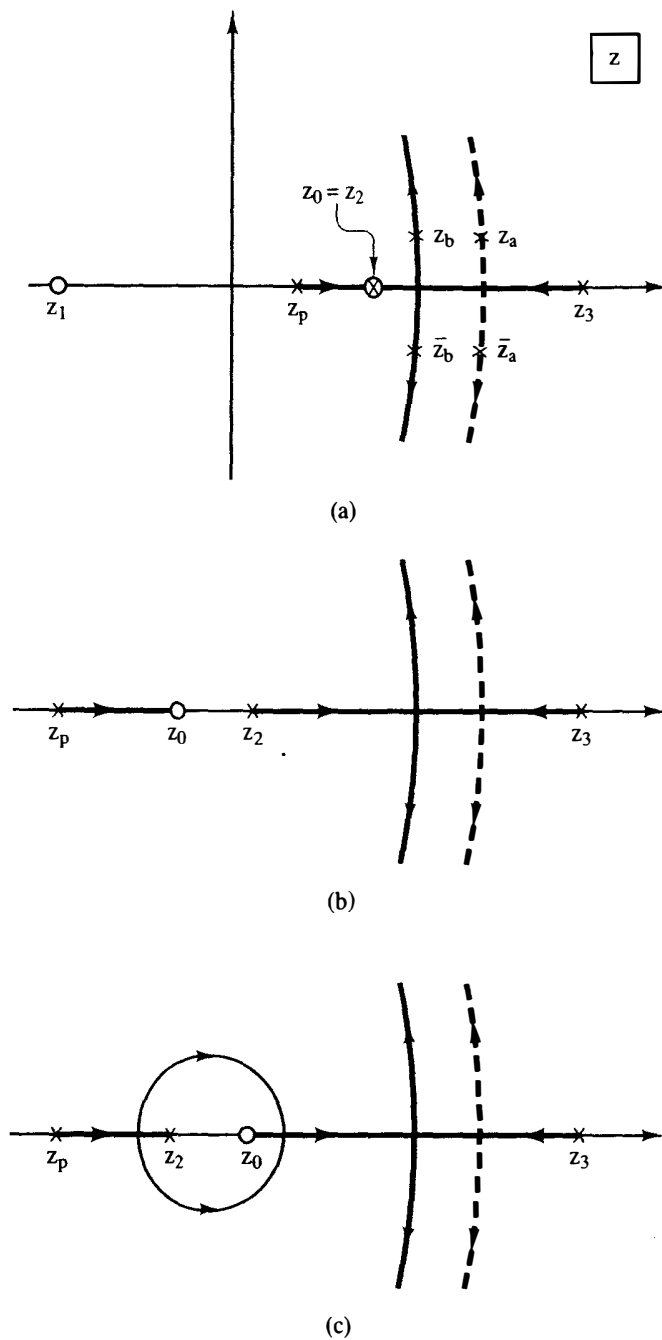


Figure 8-29 Phase-lead design.

2. Choose the compensator zero z_0 to cancel a pole of $G(z)$.
3. Choose either the gain K_c of the compensated system or the location of the compensator pole $z_p < z_0$, such that the compensator is phase lead.
4. Then, in (8-65),

$$K_c D(z_b) G(z_b) = -1 \quad (8-76)$$

Solve for the unknown, which is either K_c or z_p . The pole z_p appears twice in (8-76), as seen from (8-71) and (8-72).

Note that this design procedure sets only one root of the characteristic equation; the locations of the other roots may not be satisfactory. Hence iterations of steps 3 and 4 may be required. As in the case of frequency-response design, phase-lead design tends to be by trial and error.

As a practical consideration, exact pole-zero cancellation cannot occur. Hence, we actually have two different cases. The first case is illustrated in Figure 8-29b, where the compensator zero is to the left of the plant pole to be canceled. For this case, the closed-loop transfer function will have an additional pole slightly to the left of the compensator zero. For the second case, the compensator zero is to the right of the plant pole, as illustrated in Figure 8-29c. The closed-loop transfer function will then have an additional pole slightly to the right of the compensator zero. In either case, the amplitude of the transient-response term associated with this added closed-loop pole, as excited by the input, will be small, since the closed-loop transfer function has a zero (from the compensator) that is almost coincident with the pole.

In summary, the phase-lag controller shifts the root locus very little, but allows a higher open-loop gain to be used. Or if the same open-loop gain is used, the system is more stable. The phase-lead controller shifts the root locus to the left, resulting in a system that responds faster. A phase-lead example will now be given.

Example 8.7

Phase-lead design by trial and error will be illustrated in this example. Consider the system in Figure 8-28. Suppose that the plant transfer function is given by

$$G_p(s) = \frac{1}{s(s+1)}$$

Since the time constant of the pole at $s = -1$ is 1 s, we choose $T = 0.1$ s. Then

$$\begin{aligned} G(z) &= \frac{z-1}{z} \mathcal{Z} \left[\frac{K}{s^2(s+1)} \right] \\ &= \frac{0.004837K(z+0.9672)}{(z-1)(z-0.9048)} \end{aligned}$$

The root locus for $G(z)$ is shown in Figure 8-30. Note that K is equal to 0.244 for critical damping, with the two roots coincident at $z = 0.952$. We will choose a phase-lead controller with the zero at 0.9048, in order to cancel one of the plant poles. We will place the controller pole at $z = 0.7$, which should increase the system speed of response. Then

$$D(z) = \frac{K_d(z-z_0)}{z-z_p} = \frac{3.15(z-0.9048)}{z-0.7}$$

Note that $K_d = 3.15$ such that $D(1) = 1$. The root locus of the compensated system is also shown in Figure 8-30. We choose critical damping as our design criterion; a value of $K = 0.814$ results in a critically damped system, with roots at $z = 0.844$; these values were found by calculating the breakaway points. A pole in the s -plane at $s = -a$ has a time constant of $\tau = 1/a$ and an equivalent z -plane location of $e^{-aT} = e^{-T/\tau}$. Thus, for the uncompensated critically damped case,

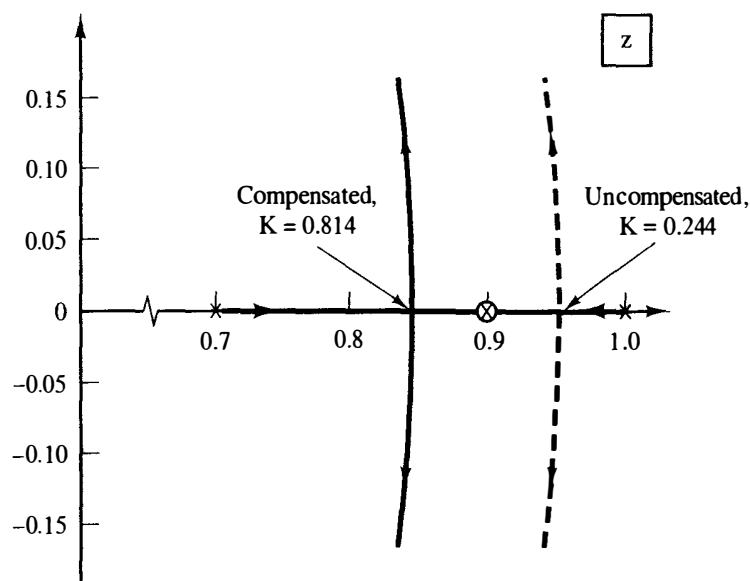


Figure 8-30 Root locus for Example 8.7.

$$e^{-0.1/\tau} = 0.952$$

or $\tau = 2.03$ s. For the compensated critically damped case,

$$e^{-0.1/\tau} = 0.844$$

or $\tau = 0.59$ s. Thus the compensated system responds much faster. A plot giving the step responses for both the uncompensated system and the compensated system is shown in Figure 8-31.

A point must be made concerning the last example. It appears that we can increase the speed of response of this system by a very large factor, simply by moving

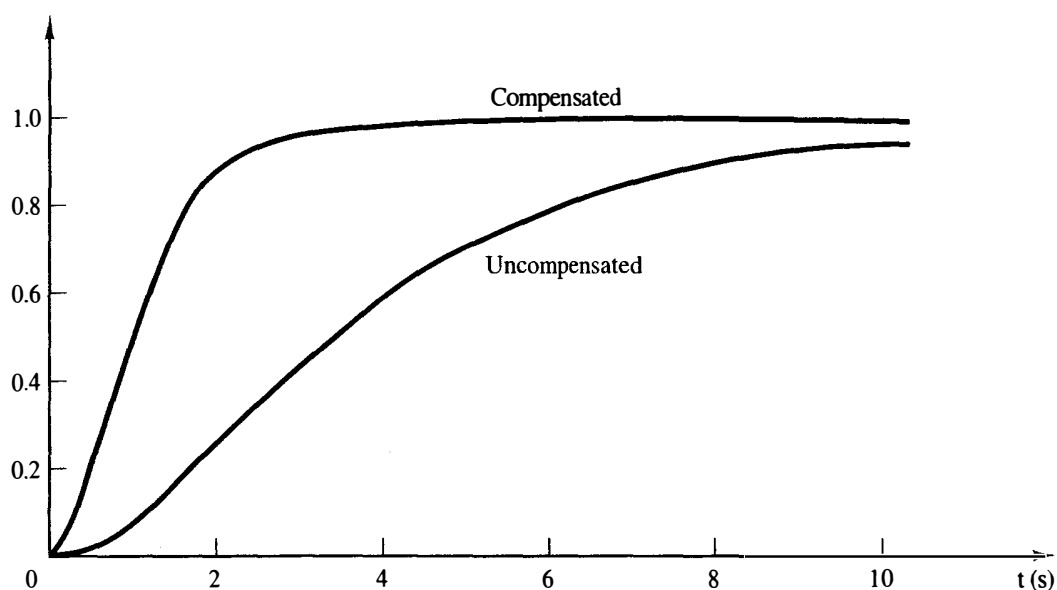


Figure 8-31 Step responses for Example 8.7.

the pole of the controller to the left. However, this movement of the pole increases the high-frequency gain of the controller, and this increase may not be acceptable. To illustrate this point, suppose that the plant in Figure 8-28 is a servomotor, and the gain K is a power amplifier that drives the motor. For the uncompensated system with a unit-step input, the maximum error is 1, which appears at $t = 0$ as shown in Figure 8-31. Since the amplifier gain $K = 0.244$, the maximum amplifier output is 0.244. For the compensated system, the maximum error is also 1, at $t = 0$. At this instant, the controller output is 3.15 (the interested reader can prove this). Hence, since the amplifier gain is 0.814, the amplifier output is 2.56, which is an increase by more than a factor of 10 over that of the uncompensated system. This increase in motor voltage accounts for the faster response. However, the larger voltages generated by the phase-lead compensation may force the system into nonlinear regions of operation. A discussion of the possible effects of this operation is beyond the scope of this book; however, the results of the linear analysis given above will no longer be applicable.

8.12 SUMMARY

Various criteria used in the specification of control systems are presented in this chapter. Next, digital controller design techniques using phase-lead and phase-lag compensation are developed. These techniques, based on frequency responses, tend to be largely trial and error, but are among the most commonly used techniques in compensator design. Then the three-term controller (PID) is developed, and is seen to be a special type of lag-lead design. Finally, design by root-locus techniques is developed.

REFERENCES AND FURTHER READING

1. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2d ed. Englewood Cliffs, NJ: Prentice Hall, 1991.
2. M. E. Van Valkenberg, *Network Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1974.
3. K. Ogata, *Modern Control Engineering*, 2nd ed. New York: McGraw-Hill Book Company, 1990.
5. M. L. Dertouzos, M. Athans, R. N. Spann, and S. J. Mason, *Systems, Networks, and Computation: Basic Concepts*. New York: McGraw-Hill Book Company, 1972.
6. C. I. Huang, "Computer Aided Design of Digital Controllers," M.S. thesis, Auburn University, Auburn, AL, 1981.
7. W. R. Wakeland, "Bode Compensator Design," *IEEE Trans. Autom. Control*, Vol. AC-21, p. 771, Oct. 1976.

8. J. R. Mitchell, "Comments on Bode Compensator Design," *IEEE Trans. Autom. Control*, Vol. AC-22, p. 869, Oct. 1977.
9. B. C. Kuo, *Digital Control Systems*, 2d ed. New York: Saunders College Publishing, 1992.
10. J. A. Cadzow and H. R. Martens, *Discrete-Time and Computer Control Systems*. Englewood Cliffs, NJ: Prentice Hall, 1970.
11. R. C. Dorf, *Modern Control Systems*, 5th ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1991.

PROBLEMS

- 8-1. Consider the second-order analog system described by

$$T(s) = \frac{G_p(s)}{1 + G_p(s)}, \quad G_p(s) = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)}$$

where $T(s)$ is the closed-loop transfer function and $G_p(s)$ is the plant transfer function. Show that the phase margin ϕ_m and the damping ratio ζ are related by $\phi_m \approx 100\zeta$ by sketching this relationship onto Figure 8-2.

- 8-2. Consider the system of Figure P8-2. The plant frequency response $G(j\omega_w)$ is given in Table P8-2.

TABLE P8-2 FREQUENCY RESPONSE FOR PROBLEM 8-2

ω_w	ω	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.099	9.95473	19.96	-98.56
0.2	0.199	4.91192	13.82	-106.98
0.3	0.297	3.20693	10.12	-115.10
0.4	0.394	2.34103	7.38	-122.83
0.5	0.490	1.81474	5.17	-130.10
0.6	0.582	1.46124	3.29	-136.87
0.7	0.673	1.20853	1.64	-143.14
0.8	0.761	1.02011	0.17	-148.92
0.9	0.845	0.87530	-1.15	-154.24
1.0	0.927	0.76140	-2.36	-159.13
2.0	1.570	0.30058	-10.44	-190.88
3.0	1.965	0.18220	-14.78	-205.37
4.0	2.214	0.13244	-17.55	-212.26
5.0	2.380	0.10579	-19.51	-215.43
6.0	2.498	0.08942	-20.97	-216.61
7.0	2.585	0.07848	-22.10	-216.68
8.0	2.651	0.07073	-23.00	-216.11
9.0	2.704	0.06502	-23.73	-215.18
10.0	2.746	0.06068	-24.33	-214.06
20.0	2.942	0.04455	-27.02	-203.02
30.0	3.008	0.04097	-27.75	-196.54
40.0	3.041	0.03964	-28.03	-192.77

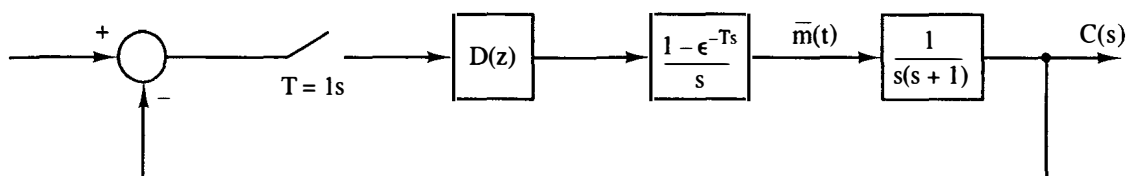


Figure P8-2 System for Problem 8-2.

- (a) Let $D(z) = 1$. If the system is stable, find its phase margin.
 - (b) Let the digital controller realize a gain K , that is, $D(z) = K$. Find K such that the system phase margin is 45° .
 - (c) Obtain the system step response of part (b) by simulation, and find the rise time t_r and the percent overshoot.
 - (d) Design a unity-dc-gain phase-lag compensator that yields a phase margin of approximately 45° .
 - (e) Obtain the system step response of part (d) by simulation, and find the rise time t_r and the percent overshoot.
- 8-3. Consider the system of Problem 8-2.
- (a) Design a unity-dc-gain phase-lead compensator that yields a phase margin of approximately 45° .
 - (b) Obtain the system step response of part (a) by simulation, and find the rise time t_r and the percent overshoot.
 - (c) Compare the results of simulation of part (b) with Problem 8-2(c) and (e).
- 8-4. Consider the system of Figure P8-2 but with $T = 0.2$.
- (a) Show that the pulse transfer function of the plant is given by

$$G(z) = \frac{z-1}{z} \mathcal{Z} \left[\frac{1}{s^2(s+1)} \right] = \frac{0.01873z + 0.01752}{(z-1)(z-0.8187)}$$
 - (b) The frequency response for $G(z)$ is given in Table P8-4. From this frequency response, sketch the Bode and the Nyquist diagrams for the uncompensated system, indicating the gain and phase margins.
 - (c) Calculate the value of $G(j\omega_w)$ as $\omega_w \rightarrow \infty$. It is not necessary to find $G(w)$ to calculate this value.
 - (d) Use the results of part (b) to estimate the overshoot in the unit-step response.
 - (e) Based on the results in part (d), are the zeros of the uncompensated system characteristic equation real or complex? Why?
 - (f) Simulate the system to check the results in part (e). Also, find the rise time t_r . The simulation will show a 21 percent overshoot.
- 8-5. Consider the system of Problem 8-4. It is desired that the steady-state error constant for a unit-ramp input, K_v of (6-20), be 4, resulting in a steady-state error of 0.25.
- (a) Repeat Problem 8-4(a), (b), (c), and (d) for this system.
 - (b) Use the phase margin ϕ_m and Figure 8-2 to estimate the characteristics ζ , M_r , and M_p .
 - (c) If simulation facilities are available, verify the value of M_p in part (b). The simulation will show that $M_p \approx 1.62$.
- 8.6. To satisfy the steady-state constraints for the system of Problem 8-5, the dc gain of the digital controller must be equal to 4.

TABLE P8-4 FREQUENCY RESPONSE FOR PROBLEM 8-4

ω_w	ω	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	9.95054	19.95	-96.28
0.2	0.200	4.90325	13.80	-102.45
0.3	0.299	3.19331	10.08	-108.41
0.4	0.399	2.32198	7.31	-114.08
0.5	0.499	1.78990	5.05	-119.40
0.6	0.599	1.43046	3.10	-124.36
0.7	0.698	1.17191	1.37	-128.95
0.8	0.798	0.97794	-0.19	-133.17
0.9	0.897	0.82799	-1.63	-137.05
1.0	0.996	0.70945	-2.98	-140.61
2.0	1.974	0.22743	-12.86	-164.43
3.0	2.914	0.10973	-19.19	-177.74
4.0	3.805	0.06511	-23.72	-187.04
5.0	4.636	0.04372	-27.18	-194.33
6.0	5.404	0.03186	-29.93	-200.38
7.0	6.107	0.02459	-32.18	-205.55
8.0	6.747	0.01980	-34.06	-210.03
9.0	7.328	0.01646	-35.67	-213.95
10.0	7.854	0.01403	-37.05	-217.40
20.0	11.071	0.00558	-45.07	-236.77
30.0	12.490	0.00352	-49.07	-243.95
40.0	13.258	0.00259	-51.73	-246.94

- (a) Design a phase-lag controller with this dc gain that will result in a 45° phase margin.
- (b) Estimate the percentage of overshoot in the step response of the compensated system.
- (c) If simulation facilities are available, verify the step-response results in part (b). The simulation will show an overshoot in the step response of approximately 24 percent.
- 8-7.** Repeat Problem 8-6 using a phase-lead controller. In part (c), the overshoot is approximately 26 percent.
- 8-8.** Consider the system of Problem 8-4. The plant frequency response $G(j\omega_w)$ is given in Table P8-4.
- (a) Let the digital controller realize a gain K , that is, $D(z) = K$. Find K such that the system phase margin is 45° .
- (b) Obtain the system step response of part (a) by simulation, and find the rise time t_r and the percent overshoot.
- 8-9.** Three constraints are given on the choice of the phase-margin frequency, ω_{w1} , in Section 8.6 for phase-lead design. Derive the three equivalent constraints on ω_{w1} for the design of phase-lag controllers using (8-33).
- 8-10.** Repeat Problem 8-2, using a proportional-plus-integral digital filter.
- 8-11.** Repeat Problem 8-3, using a proportional-plus-derivative filter.
- 8-12.** Consider Problem 8-6 again. In this problem a proportional-plus-integral (PI) filter is to be designed.

- (a) The gain added in Problem 8-6 to reduce steady-state errors is no longer necessary. Why?
- (b) Repeat Problem 8-6 using a proportional-plus-integral filter, with the gain of 4 omitted.
- 8-13. Repeat Problem 8-7 using a proportional-plus-derivative (PD) filter.
- 8-14. Shown in Figure P8-14 is the block diagram of the temperature control system described in Problem 1-10. For this problem, ignore the disturbance input. In Figure P8-14 the sensor gain of $H = 0.04$ has been shifted into the forward path to yield a unity feedback system. The stability characteristics are unchanged, since the loop transfer function has not changed. Note that $c(t)$ is the chamber temperature in degrees Celsius. It is shown in Problem 6-4 that with $T = 0.6$, the function $G(z)H$ is given by

$$G_e(z) = (0.04) \frac{z-1}{z} \mathcal{Z} \left[\frac{2}{s(s+0.5)} \right] = \frac{0.04147}{z-0.7408}$$

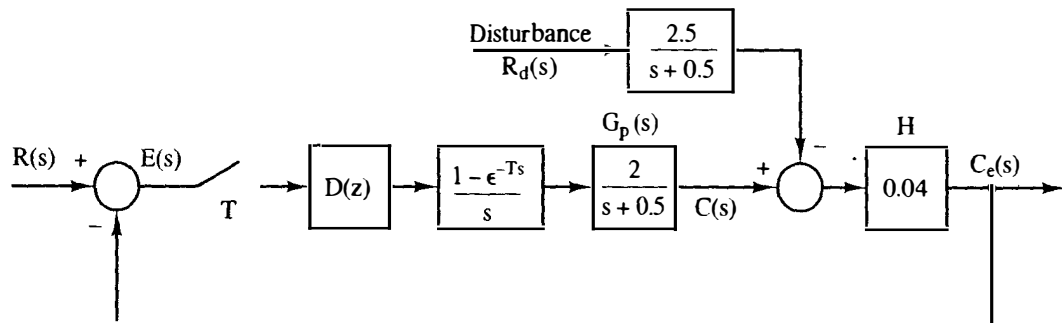


Figure P8-14 Modified temperature control system.

- The frequency response for $G(z)$ is given in Table P8-14. The sensor gain $H = 0.04$ is *not* included in this table.
- (a) Calculate the dc gains of both the forward path and the closed-loop system, with $D(z) = 1$.
- (b) Find the steady-state error $e_{ss}(kT)$ for a unit step input using (6-19). Express this error in percent.
- (c) Let $D(z) = K$, a pure gain. Find K such that the steady-state error in part (b) is 0.05, or 5 percent. Assume that the resulting system is stable.
- (d) It was shown in Problem 7-9 that the system is unstable for $K > 41.96$. Can the system in part (c) be implemented to yield a 5 percent steady-state error? Explain your answer.
- (e) Design a phase-lag compensator with a dc gain of 118.8 to yield a phase margin of 45° .
- (f) What is the percent steady-state error for a constant input for the system in part (e)?
- (g) If simulation facilities are available, simulate the system of part (e) to show that the steady-state error is 5 percent.
- 8-15. This problem is based on the solution in Problem 8-14. Suppose that in Figure P8-14, the sensor gain $H = 0.04$ is moved back to the feedback path. What effect does this have on the percent steady-state error of Problem 8-14, where this error is defined as

TABLE P8-14 FREQUENCY RESPONSE FOR PROBLEM 8-14

ω_w	ω	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	3.92295	11.87	-13.11
0.2	0.199	3.71673	11.40	-25.38
0.3	0.299	3.43700	10.72	-36.29
0.4	0.398	3.13670	9.92	-45.71
0.5	0.496	2.84938	9.09	-53.74
0.6	0.593	2.59043	8.26	-60.60
0.7	0.690	2.36393	7.47	-66.52
0.8	0.785	2.16851	6.72	-71.68
0.9	0.879	2.00067	6.02	-76.23
1.0	0.971	1.85649	5.37	-80.30
2.0	1.801	1.12345	1.01	-107.02
3.0	2.442	0.87830	-1.12	-122.59
4.0	2.920	0.76932	-2.27	-133.12
5.0	3.276	0.71225	-2.94	-140.64
6.0	3.545	0.67895	-3.36	-146.21
7.0	3.754	0.65796	-3.63	-150.48
8.0	3.920	0.64393	-3.82	-153.83
9.0	4.053	0.63411	-3.95	-156.52
10.0	4.163	0.62698	-4.05	-158.72
20.0	4.685	0.60357	-4.38	-169.11
30.0	4.867	0.59912	-4.44	-172.71
40.0	4.958	0.59756	-4.47	-174.52

$$\text{steady-state error} = \frac{\text{commanded output} - \text{actual output}}{\text{commanded output}} \times 100$$

Note that this result allows us to convert a nonunity feedback gain system to a unity-gain feedback system, without affecting the percentage steady-state error.

- 8-16.** Repeat Problem 8-14 for a proportional-plus-integral (PI) compensator. Note that in this case, the steady-state error for a constant input is zero.
- 8-17.** In this problem we consider the effects of the disturbance input in Figure P8-14.
- Suppose that the disturbance input is a unit step, which models the door to the chamber remaining open. Find the steady-state error in the temperature $c(t)$ for the uncompensated system in Problem 8-14.
 - Repeat part (a) for the phase-lag compensated system in Problem 8-14(e).
 - Repeat part (a) for the PI-compensated system in Problem 8-16.
- 8-18.** Consider the block diagram of a robot-arm control system shown in Figure P8-18. This system is described in Problem 1-16. Let $T = 0.1$. It was shown in Problem 4-15 that

$$G(z) = \frac{z-1}{z} \left[\frac{2}{s^2(0.5s+1)} \right] = \frac{0.01873z + 0.01752}{(z-1)(z-0.8187)}$$

The frequency response for $G(z)$ is given in Table P7-23. Note that the sensor gain $H = 0.07$ is not included in this table.

- Find the system phase margin with $D(z) = 1$.

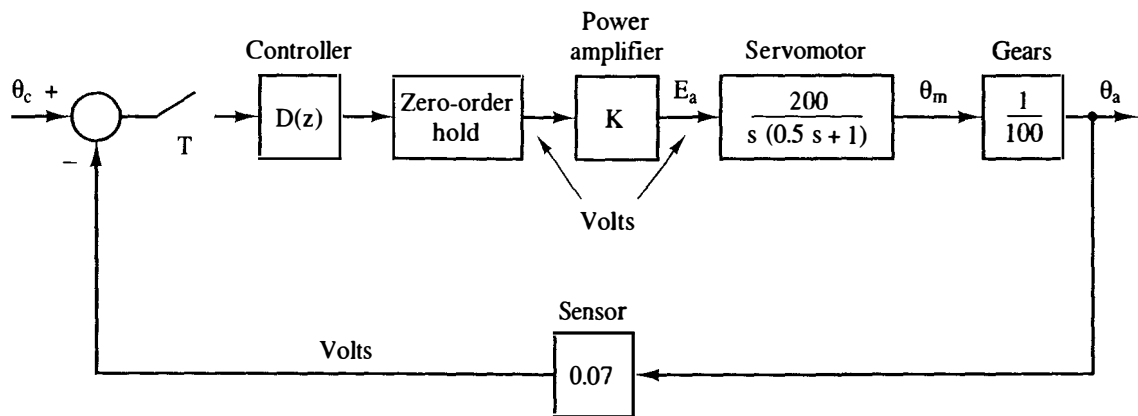


Figure P8-18 Robot arm joint control system.

- (b) Design a phase-lag controller with the dc gain of 10 that yields a system phase margin of 45° .
- (c) Design a phase-lead controller with the dc gain of 10 that yields a system phase margin of 45° .
- (d) If simulation facilities are available, find the step response for the systems of parts (b) and (c), with $\theta_c(t) = 0.07u(t)$. Compare the rise times and the percent overshoot for the two systems. The percent overshoot is defined as

$$\text{percent overshoot} = \frac{\text{maximum value} - \text{final value}}{\text{final value}} \times 100$$

- 8-19. (a) Design a PI controller for Problem 8-18(b).
 (b) Design a PD controller for Problem 8-18(c).
 (c) Use the results of parts (a) and (b) to repeat Problem 8-18(d).
- 8-20. Consider the block diagram of an antenna control system shown in Figure P8-20. This system is described in Section 1.5. Let $T = 0.05$ and the sensor gain be unity ($H = 1$).

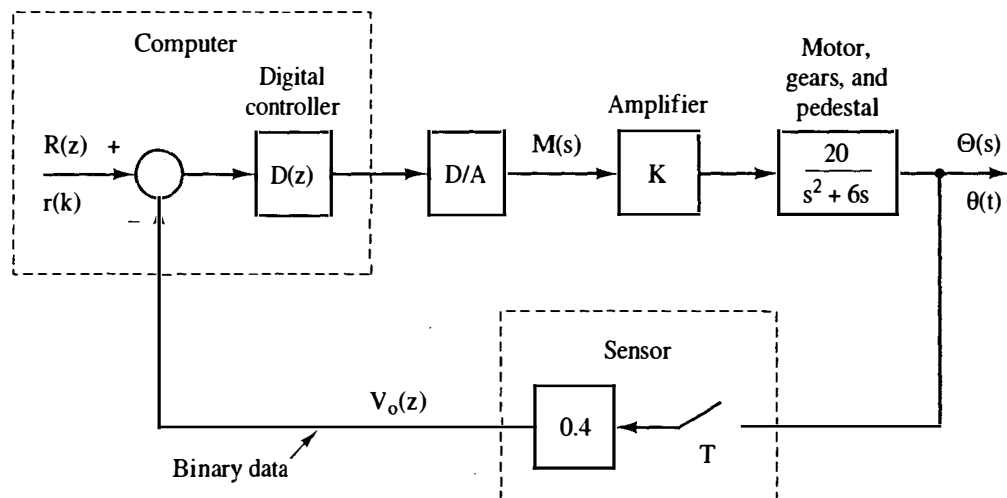


Figure P8-20 Block diagram for an antenna control system.

It was shown in Problem 7-11 that

$$G(z) = \frac{z-1}{z} \mathcal{Z} \left[\frac{20}{s^2(s+6)} \right] = \frac{0.02268z + 0.02052}{(z-1)(z-0.7408)}$$

The frequency response for $G(z)$ is given in Table P7-24.

- (a) Find the system phase margin with $K = 1$ and $D(z) = 1$.
 - (b) To reduce steady-state errors, K is increased to 5. Design a unity-dc-gain phase-lag controller that yields a system phase margin of 45° .
 - (c) Design a unity-dc-gain phase-lead controller, with $K = 5$, that yields a system phase margin of 45° .
 - (d) If simulation facilities are available, find the unit step response for the systems of parts (b) and (c). Compare the rise times and the percent overshoot for the two systems.
- 8-21.** (a) Design a PI controller for Problem 8-20(b).
 (b) Design a PD controller for Problem 8-20(c).
 (c) Use the results of parts (a) and (b) to repeat Problem 8-20(d).
- 8-22.** Consider the block diagram of a satellite control system shown in Figure P8-22. This system is described in Problem 1-12. Let $T = 0.1$ s, $K = 1$, $J = 0.1$, and $H_k = 0.02$. It was shown in Problem 7-12 that

$$G(z) = \frac{z-1}{z} \mathcal{Z} \left[\frac{10}{s^3} \right] = \frac{0.05(z+1)}{(z-1)^2}$$

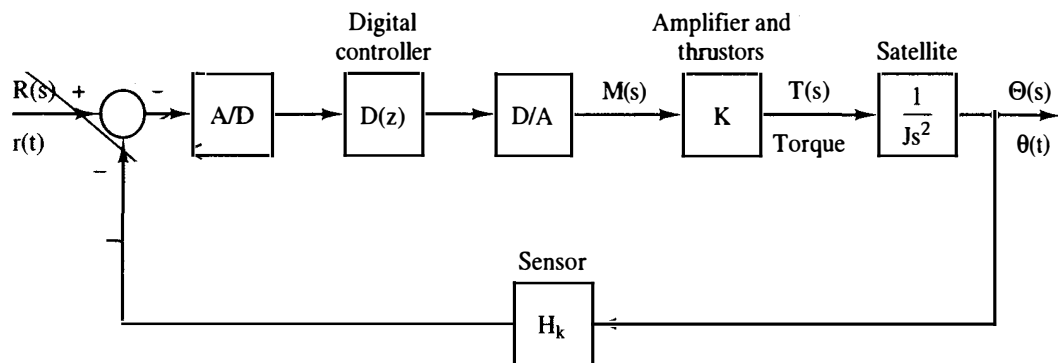


Figure P8-22 Block diagram for a satellite control system.

The frequency response of $G(z)$ is given in Table P7-25. Note that the gain $H_k = 0.02$ is not included in this table.

- (a) Sketch the complete Nyquist diagram for $D(z) = 1$. Is this system stable?
 - (b) This system cannot be stabilized by any type of phase-lag controller. Why? *Hint:* Consider the Nyquist criterion and the Nyquist diagram in part (a).
 - (c) Design a unity-dc-gain phase-lead controller that yields a system phase margin of 45° . *Hint:* Use $\omega_{w1} = 1$.
 - (d) If simulation facilities are available, find the unit step response for the system of part (c). Find the rise time and the percent overshoot for this system.
- 8-23.** (a) Repeat Problem 8-22(a) and (b).
 (b) Design a PD controller for Problem 8-22(c). *Hint:* Use $\omega_{w1} = 1$.
 (c) Use the results of part (b) to repeat Problem 8-22(d).

- 8-24. Write the difference equations required for the realization of the PID controller transfer function $D(z)$ in (8-52). Let $E(z)$ be the controller input and $M(z)$ be the controller output.
- 8-25. If the rectangular integrator [see below (8-38)] is used in the PID transfer function in (8-52), the resulting controller is described by

$$\frac{M(z)}{E(z)} = D(z) = K_p + K_I \left[\frac{Tz}{z-1} \right] + K_D \left[\frac{z-1}{Tz} \right]$$

This transfer function is implemented in many commercial digital controllers. Write the difference equation required to realize this controller.

- 8-26. Consider the system of Example 8.1.
- Design a PI filter to achieve a phase margin of 60° .
 - Obtain the system step response by simulation. Compare this response to that of the system of Example 8.1, which is plotted in Figure 8-14.
- 8-27. Consider the chamber temperature control system of Problem 8-14. Suppose that a variable gain K is added to the plant. The pulse transfer function for this system is given in Problem 8-14.
- Plot the root locus for this system and find the value of $K > 0$ for which the system is stable.
 - Find the time constant for the system with $K = 1$.
 - Design a phase-lag controller such that the characteristic-equation root is approximately the same as that in part (b) but for the gain $K = 3$. *Hint:* Choose $z_p = 0.999$.
 - Find the steady-state errors for both the uncompensated system of part (b) and the compensated system of part (c).
 - By computer, verify the design in part (c) by finding the characteristic-equation roots.
- 8-28. Consider the chamber temperature control system of Problem 8-14. Suppose that a variable gain K is added to the plant. The pulse transfer function for this system is given in Problem 8-14.
- Plot the root locus for this system, and find the value of $K > 0$ for which the system is stable.
 - Find the time constant for the system with $K = 1$.
 - Assuming the resulting system to be stable, find K such that the steady-state error in the uncompensated system is 0.05, or 5 percent. Is the resulting system stable? [See part (a).]
 - Design a phase-lag controller such that the characteristic-equation root is approximately the same for the gain K found in part (c). *Hint:* Choose $z_p = 0.99999$.
 - By computer, verify the design in part (d) by finding the characteristic-equation roots.
- 8-29. Consider the system of Problem 8-26.
- Plot the root locus for this system, and find the value of $K > 0$ for which the system is stable.
 - Find the time constant for the system with $K = 1$.
 - Design a phase-lead controller such that the characteristic-equation root in part (b) moves to yield a time constant of 0.5 s for $K = 1$.
 - By computer, verify the design in part (c) yields the correct characteristic-equation zero.

8-30. Consider the system of Figure P8-2 with a gain factor K added to the plant.

(a) Show that the pulse transfer function is given by

$$\frac{z-1}{z} \mathcal{Z} \left[\frac{1}{s^2(s+1)} \right] = \frac{0.368(z+0.717)}{(z-1)(z-0.368)}$$

(b) Sketch the root locus for this system, and find the value of K that results in critical damping, that is, in two real and equal roots of the system characteristic equation.

(c) Find the time constants of the two poles in part (b).

(d) It is desired that the gain in part (b) be multiplied by 1.8, while the characteristic-equation roots remain approximately the same. Design a phase-lag controller that meets these specifications. *Hint:* Use $z_p = 0.999$.

(e) By computer, verify the characteristic-equation roots in part (d).

8-31. Consider the system of Example 8.7.

(a) Design a phase-lag compensator such that with $K = 0.5$, the system is critically damped with roots having a time constant of approximately 2.03 s. This time constant is the same as that of the uncompensated system with $K = 0.244$.

(b) By computer, verify the characteristic-equation roots in part (a).

Pole-Assignment Design and State Estimation

9.1 INTRODUCTION

In Chapter 8 we considered discrete control design both from a frequency-response point of view and from a root-locus point of view. In every case we considered that only one signal was to be fed back. It seems reasonable that if we determine more about the present condition of a system, and use this additional information to generate the control input, then we should be able to control the system in a manner that is, in some sense, better. Of course, we have a complete description of the condition of a system if we measure its state vector. Hence we might conclude that if we are able to specify mathematically what defines the very best control system, the design of this control system might require that we have all the states of the system available for feedback. Generally, the design of systems of this type does require that we assume that the full state vector is available for feedback.

For most control systems the measurement of the full-state vector is impractical. To implement a design based on full state feedback, we must estimate the states of a system using measurements that are practical. The state estimation is accomplished by designing a dynamic system (a set of equations) for the computer that estimates the states using all information available. Fortunately, we can separate the design into two phases. During the first phase, we design the system as though all states of the system will be measured. The second phase is then the design of the state estimator. In this chapter we consider both phases of the design process and the effects that the state estimator has on closed-loop system operation. Then, in Chapter 10, after the introduction of the appropriate mathematics, we consider the Kalman filter, which is an optimal state estimator.

It is obvious from the calculations in this chapter that most designs require computer calculations. The computer programs CTRL and CSP, described in Appendix VI, implement the design procedures of this chapter. These programs can be used to verify the examples and the solutions of most of the problems in this chapter.

9.2 POLE ASSIGNMENT

In this section a design procedure generally known as pole assignment, or pole placement, is developed. The design results in the assignment of the poles of the closed-loop transfer function (zeros of the characteristic equation) to any desired locations. There are, of course, practical implications that will be discussed as the technique is developed.

To introduce the pole-assignment technique, we will consider the model of a servomotor, developed in Chapter 1. This model is second-order, and an example is shown in Figure 9-1. The state model for this system was calculated in Section 4.10 and is given by

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned} \quad (9-1)$$

In this model $x_1(k)$ is the position (angle) of the motor shaft, which can easily be measured. The state $x_2(k)$ is shaft velocity, which can be measured using a tachometer or some other suitable sensor. Thus for this system the full state vector can be measured.

We choose the control input $u(k)$ to be a linear combination of the states; that is,

$$u(k) = -K_1 x_1(k) - K_2 x_2(k) = -\mathbf{K} \mathbf{x}(k) \quad (9-2)$$

where the gain matrix \mathbf{K} is

$$\mathbf{K} = [K_1 \quad K_2]$$

Then (9-1) can be written as

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) - \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} [K_1 x_1(k) + K_2 x_2(k)] \\ &= \begin{bmatrix} 1 - 0.00484K_1 & 0.0952 - 0.00484K_2 \\ -0.0952K_1 & 0.905 - 0.0952K_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \end{aligned} \quad (9-3)$$

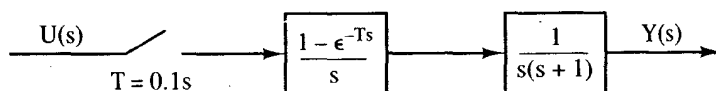


Figure 9-1 Servomotor system.

$$A - BK = A_c$$

The matrix of (9-3) is the closed-loop system matrix, which we will call A_f . The closed-loop system equation is then

$$\mathbf{x}(k+1) = A_f \mathbf{x}(k) \quad (9-4)$$

and the characteristic equation is

$$|z\mathbf{I} - A_f| = 0 \quad (9-5)$$

Evaluation of (9-5) yields, after some calculations, the characteristic equation

$$z^2 + (0.00484K_1 + 0.0952K_2 - 1.905)z + 0.00468K_1 - 0.0952K_2 + 0.905 = 0 \quad (9-6)$$

Suppose that, by some process we choose the desired characteristic-equation zero locations to be λ_1 and λ_2 . Then the desired characteristic polynomial, denoted $\alpha_c(z)$, is given by

$$\alpha_c(z) = (z - \lambda_1)(z - \lambda_2) = z^2 - (\lambda_1 + \lambda_2)z + \lambda_1\lambda_2 \quad (9-7)$$

Equating coefficients in (9-6) and (9-7) yields the equations

$$\begin{aligned} 0.00484K_1 + 0.0952K_2 &= -(\lambda_1 + \lambda_2) + 1.905 \\ 0.00468K_1 - 0.0952K_2 &= \lambda_1\lambda_2 - 0.905 \end{aligned} \quad (9-8)$$

These equations are linear in K_1 and K_2 , and upon solving yield

$$\begin{aligned} K_1 &= 105[\lambda_1\lambda_2 - (\lambda_1 + \lambda_2) + 1.0] \\ K_2 &= 14.67 - 5.34\lambda_1\lambda_2 - 5.17(\lambda_1 + \lambda_2) \end{aligned} \quad (9-9)$$

Thus we can find the gain matrix \mathbf{K} that will realize *any* desired characteristic equation.

We wish now to make the following points. By some process we choose the root locations so as to satisfy design criteria such as speed of response, overshoot in the transient response, and the like (see Section 6.4). However, at this time no input has been shown; hence we cannot speak of overshoot to a step input. Once we have chosen λ_1 and λ_2 , (9-9) will give the gain matrix needed to realize these characteristic-equation zeros. Then, to the extent that (9-1) is an accurate model of the physical servomotor, we will realize these zeros in the physical system. Obviously, we cannot increase the speed of response of a servo system without limit, even though (9-9) seems to indicate that we can. If we attempt to force the system to respond too rapidly, large signals will be generated and the plant will enter a nonlinear mode of operation; then (9-1) will no longer accurately model the plant. Hence, in choosing λ_1 and λ_2 , we must consider only those root locations that can be attained by the physical system.

We will now develop a general procedure of pole assignment. Our n th-order plant is modeled by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (9-10)$$

We generate the control input $u(k)$ by the relationship

$$u(k) = -\mathbf{K}\mathbf{x}(k) \quad (9-11)$$

where

$$\mathbf{K} = [K_1 \quad K_2 \quad \cdots \quad K_n] \quad (9-12)$$

Then (9-10) can be written as

$$\mathbf{x}(k+1) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(k) \quad (9-13)$$

and, in (9-4), $\mathbf{A}_f = (\mathbf{A} - \mathbf{BK})$. We choose the desired pole locations

$$z = \lambda_1, \lambda_2, \dots, \lambda_n \quad (9-14)$$

Then the closed-loop system characteristic polynomial is

$$\alpha_c(z) = |z\mathbf{I} - \mathbf{A} + \mathbf{BK}| = (z - \lambda_1)(z - \lambda_2) \cdots (z - \lambda_n) \quad (9-15)$$

In this equation there are n unknowns K_1, K_2, \dots, K_n , and n known coefficients in the right-hand-side polynomial. We can solve for the unknown gains by equating coefficients in (9-15), as illustrated by the servomotor example above. An example will now be given to illustrate this procedure.

Example 9.1

In this example the servomotor system of Figure 9-1, with the state equations given in (9-1), will be considered. With full-state feedback, the characteristic equation is given in (9-6). Consider first that the closed-loop system is implemented with the usual unity feedback; then $K_1 = 1$ and $K_2 = 0$. From (9-6), the characteristic equation is given by

$$z^2 - 1.9z + 0.91 = 0$$

This equation has roots at

$$z_{1,2} = 0.954 \angle \pm 0.091 \text{ rad} = r \angle \pm \theta$$

From (6-8) we calculate the damping factor of these roots to be

$$\zeta = \frac{-\ln r}{\sqrt{\ln^2 r + \theta^2}} = \frac{-\ln(0.954)}{\sqrt{\ln^2(0.954) + (0.091)^2}} = 0.46$$

and from (6-10), the time constant is

$$\tau = \frac{-T}{\ln r} = \frac{-0.1}{\ln(0.954)} = 2.12 \text{ s}$$

Suppose that we decide that this value of ζ is satisfactory, but that a time constant of 1.0 s is required. Then

$$\ln r = -\frac{T}{\tau} = -0.1$$

or $r = 0.905$. Solving (6-8) for θ , we have

$$\theta^2 = \frac{\ln^2 r}{\zeta^2} - \ln^2 r = \frac{\ln^2(0.905)}{(0.46)^2} - \ln^2(0.905)$$

or θ is equal to 0.193 rad, or 11.04° . Then the desired root locations are

$$\lambda_{1,2} = 0.905 / \pm 11.04^\circ = 0.888 \pm j0.173$$

Hence the desired characteristic equation is given by

$$(z - 0.888 - j0.173)(z - 0.888 + j0.173) = z^2 - 1.776z + 0.819$$

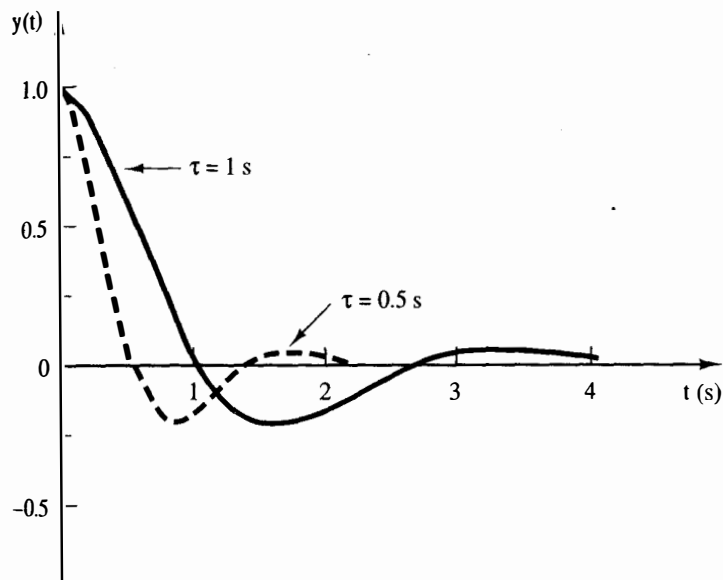
From (9-9),

$$K_1 = 105[\lambda_1 \lambda_2 - (\lambda_1 + \lambda_2) + 1.0]$$

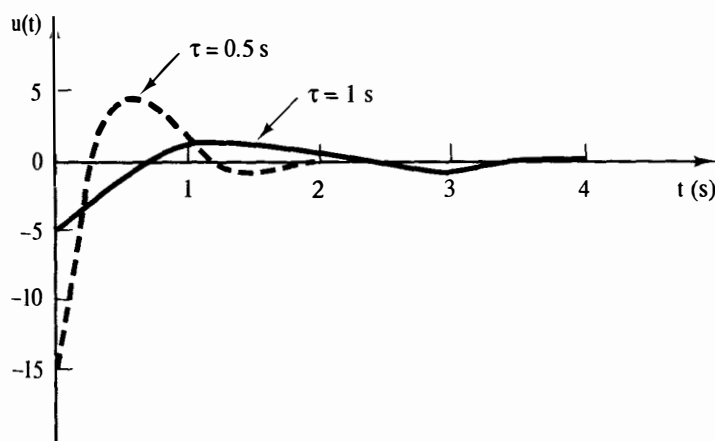
$$= 105[0.819 - (1.776) + 1.0] = 4.52$$

$$K_2 = 14.67 - 5.34\lambda_1 \lambda_2 - 5.17(\lambda_1 + \lambda_2) = 1.12$$

The initial-condition response of this system is given in Figure 9-2, with $x_1(0) = 1.0$ and



(a)



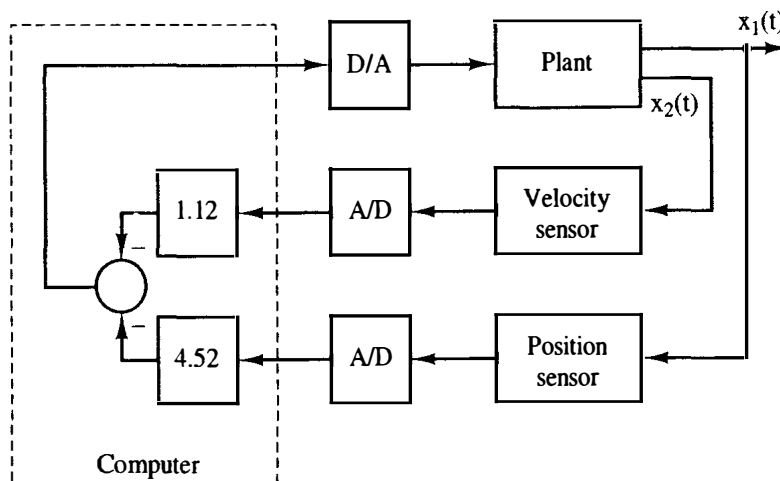
(b)

Figure 9-2 Initial condition responses for the systems of Example 9.1.

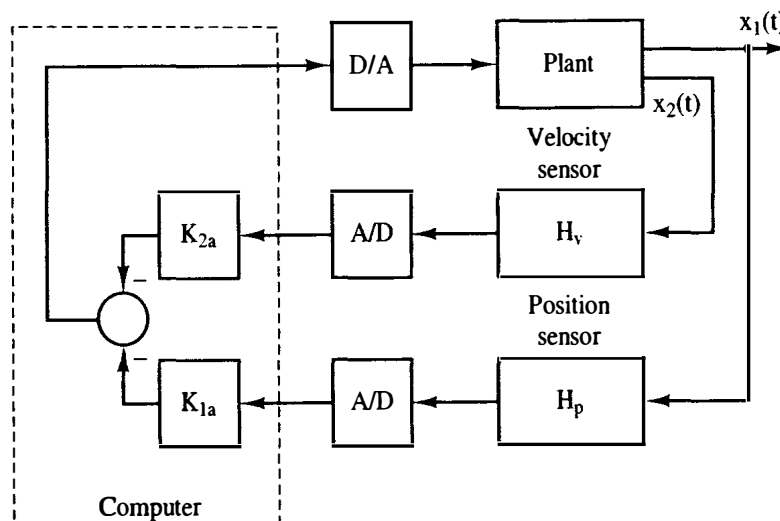
$x_2(0) = 0$. Both $y(t)$, the output (Figure 9-2a), and $u(t)$, the sampler input (Figure 9-2b), are shown.

Suppose that we decide that the foregoing value of $\zeta = 0.46$ is satisfactory, but that a time constant of $\tau = 0.5$ s is required. This design is given as Problem 9-1. The gains obtained from this design are $K_1 = 16.0$ and $K_2 = 3.26$, and the initial condition response is also given in Figure 9-2. Compare the control inputs $u(t)$ for the two designs in Figure 9-2. The maximum amplitude of $u(t)$ for $\tau = 1.0$ s is 4.52, while that for $\tau = 0.5$ s is 15.0. Thus we see that an attempt to increase the speed of response of the system results in larger signals at the plant input. These larger signals may force the system into a nonlinear region of operation, which in some cases may be undesirable.

The system of Example 9.1 can be realized with the hardware configuration of Figure 9-3. In Figure 9-3a it is assumed that the gains of the sensors are unity; in the practical case these gains would probably not be unity. For example, suppose



(a)



(b)

Figure 9-3 Hardware implementation for the design of Example 9.1.

that the sensors for position and velocity can be modeled as the constant gains H_p and H_v , respectively, over the closed-loop system bandwidth. The hardware configuration of the system would then be as shown in Figure 9-3b, where

$$H_p K_{1a} = 4.52, \quad H_v K_{2a} = 1.12$$

In the procedure above the gain matrix \mathbf{K} was calculated by equating coefficients in the characteristic equation of (9-15). This calculation is simplified greatly if the state model of the plant is in the control canonical form (see Section 2.8), which is the form

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & \ddots & \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(k) \quad (9-16)$$

This plant has the characteristic equation, from Section 2.8,

$$\alpha(z) = |z\mathbf{I} - \mathbf{A}| = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0 = 0 \quad (9-17)$$

For this state model, in (9-15) \mathbf{BK} is equal to

$$\mathbf{BK} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} [K_1 \ K_2 \ \cdots \ K_n] = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ & & \ddots & \\ K_1 & K_2 & \cdots & K_n \end{bmatrix} \quad (9-18)$$

Hence the closed-loop system matrix is

$$\mathbf{A} - \mathbf{BK} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ & & \ddots & \\ -(a_0 + K_1) & -(a_1 + K_2) & \cdots & -(a_{n-1} + K_n) \end{bmatrix} \quad (9-19)$$

and the characteristic equation becomes

$$|z\mathbf{I} - \mathbf{A} + \mathbf{BK}| = z^n + (a_{n-1} + K_n)z^{n-1} + \cdots + (a_1 + K_2)z + (a_0 + K_1) = 0 \quad (9-20)$$

If we write the desired characteristic equation as

$$\alpha_c(z) = z^n + \alpha_{n-1}z^{n-1} + \cdots + \alpha_1z + \alpha_0 = 0 \quad (9-21)$$

we calculate the gains by equating coefficients in (9-20) and (9-21) to yield

$$K_{i+1} = \alpha_i - a_i, \quad i = 0, 1, \dots, n-1 \quad (9-22)$$

In general, the techniques for developing state models for plants do not result in the control canonical form (see Section 4.10). A more practical procedure for calculating the gain matrix \mathbf{K} is the use of Ackermann's formula [1]. The proof of Ackermann's formula will not be given here (see Ref. 1 or 2), but this proof is based on transformations from a general system matrix \mathbf{A} to that of the control canonical form.

We begin with the plant model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (9-23)$$

The matrix polynomial $\alpha_c(\mathbf{A})$ is formed using the coefficients of the desired characteristic equation (9-21).

$$\alpha_c(\mathbf{A}) = \mathbf{A}^n + \alpha_{n-1}\mathbf{A}^{n-1} + \cdots + \alpha_1\mathbf{A} + \alpha_0\mathbf{I} \quad (9-24)$$

Then Ackermann's formula for the gain matrix \mathbf{K} is given by

$$\mathbf{K} = [0 \ 0 \ \cdots \ 0 \ 1][\mathbf{B} \ \mathbf{A}\mathbf{B} \ \cdots \ \mathbf{A}^{n-2}\mathbf{B} \ \mathbf{A}^{n-1}\mathbf{B}]^{-1}\alpha_c(\mathbf{A}) \quad (9-25)$$

The problems with the existence of the inverse matrix in (9-25) are discussed in Section 9.6. Example 9.1 will now be solved using Ackermann's formula.

Example 9.2



We will solve the design in Example 9.1 via Ackermann's formula. The plant model is given by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k)$$

and the desired characteristic equation is

$$\alpha_c(z) = z^2 - 1.776z + 0.819$$

Hence

$$\alpha_c(\mathbf{A}) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix}^2 - 1.776 \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} + 0.819 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

or

$$\alpha_c(\mathbf{A}) = \begin{bmatrix} 0.043 & 0.01228 \\ 0 & 0.03075 \end{bmatrix}$$

Also,

$$\mathbf{A}\mathbf{B} = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} = \begin{bmatrix} 0.0139 \\ 0.0862 \end{bmatrix}$$

Thus

$$[\mathbf{B} \ \mathbf{A}\mathbf{B}]^{-1} = \begin{bmatrix} 0.00484 & 0.0139 \\ 0.0952 & 0.0862 \end{bmatrix}^{-1} = \begin{bmatrix} -95.13 & 15.34 \\ 105.1 & -5.342 \end{bmatrix}$$

Then the gain matrix is, from (9-25),

$$\begin{aligned} \mathbf{K} &= [0 \ 1] \begin{bmatrix} -95.13 & 15.34 \\ 105.1 & -5.342 \end{bmatrix} \begin{bmatrix} 0.043 & 0.01228 \\ 0 & 0.03075 \end{bmatrix} \\ &= [4.52 \ 1.12] \end{aligned}$$

These results are the same as those obtained in Example 9.1.

A general MATLAB program to implement the calculations in this example is given by

```

order = 2;
A = [1 .0952; 0 0.905];
B = [.00484; .0952];
alphac = [1 -1.776 .819];
AMB=B;
AMBT=B;
for n=2:order
    AMBT=A*AMBT;
    for nn=1:order
        AMB(nn,n)=AMBT(nn,1);
    end
end
AMBI=inv(AMB);
CC=polyvalm(alphac,A);
CCC = AMBI*CC;
disp(' The gain matrix K is:')
disp(' ')
K=CCC(order,:);

```

9.3 STATE ESTIMATION

In Section 9.2 a design technique was developed which requires that all the plant states be measured. In general, the measurement of all the plant states is impractical, if not impossible, in all but the simplest of systems. In this section we develop a technique for estimating the states of a plant from the information that is available concerning the plant. The system that estimates the states of another system is generally called an *observer* [3], or a *state estimator*. Thus in this section we introduce the design of observers.

Suppose that the plant is described by the following equation:

$$\begin{aligned}
 \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\
 \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)
 \end{aligned}
 \tag{9-26}$$

where $\mathbf{y}(k)$ are the plant signals that will be measured. Hence, in (9-26), we know the matrices \mathbf{A} , \mathbf{B} , and \mathbf{C} , and the signals $\mathbf{y}(k)$ and $\mathbf{u}(k)$. The control inputs $\mathbf{u}(k)$ are known since we generate them. The problem of observer design can be depicted as shown in Figure 9-4, where the observer is a set of difference equations to be solved by a digital computer. The states of the system to be observed are $\mathbf{x}(k)$; the states of the observer are $\mathbf{q}(k)$ and we desire that $\mathbf{q}(k)$ be approximately equal to $\mathbf{x}(k)$. Since the observer will be implemented on a computer, the signals $\mathbf{q}(k)$ are then available for feedback calculations.

Observer Model

Equations describing observers can be developed in several different ways; we choose here to take a transfer-function approach. The observer design criterion to

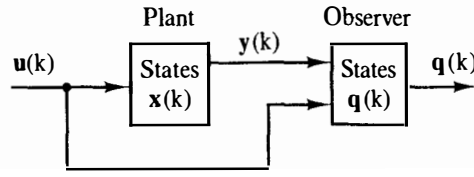


Figure 9-4 System illustrating state estimation.

be used is that the transfer-function matrix from the input $u(k)$ to the observer state $q_i(k)$ be equal to that from the input $u(k)$ to the system state $x_i(k)$, for $i = 1, 2, \dots, n$. Equations describing the observer of Figure 9-4 will now be developed, based on this reasonable criterion.

First we take the z -transform of (9-26) and solve for $X(z)$:

$$zX(z) = AX(z) + BU(z) \quad (9-27)$$

or

$$X(z) = (zI - A)^{-1} BU(z) \quad (9-28)$$

Now, since the observer has two inputs, $y(k)$ and $u(k)$, we write the observer state equations as

$$q(k+1) = Fq(k) + Gy(k) + Hu(k) \quad (9-29)$$

where the matrices F , G , and H are unknown. Taking the z -transform of (9-29) and solving for $Q(z)$ yields

$$Q(z) = (zI - F)^{-1} [GY(z) + HU(z)] \quad (9-30)$$

From (9-26),

$$Y(z) = CX(z) \quad (9-31)$$

Next we substitute (9-31) and (9-28) in (9-30):

$$\begin{aligned} Q(z) &= (zI - F)^{-1} [GCX(z) + HU(z)] \\ &= (zI - F)^{-1} [GC(zI - A)^{-1} B + H]U(z) \end{aligned} \quad (9-32)$$

Recall that the design criterion is that the transfer-function matrix from $U(z)$ to $Q(z)$ be the same as that from $U(z)$ to $X(z)$. Thus, from (9-28),

$$Q(z) = (zI - A)^{-1} BU(z) \quad (9-33)$$

must be satisfied. Then, from (9-32) and (9-33),

$$(zI - A)^{-1} B = (zI - F)^{-1} GC(zI - A)^{-1} B + (zI - F)^{-1} H \quad (9-34)$$

or

$$[I - (zI - F)^{-1} GC](zI - A)^{-1} B = (zI - F)^{-1} H \quad (9-35)$$

This equation can be expressed as

$$(zI - F)^{-1} [zI - (F + GC)](zI - A)^{-1} B = (zI - F)^{-1} H$$

Equating the postmultiplying matrices of $(z\mathbf{I} - \mathbf{F})^{-1}$ yields

$$(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = [z\mathbf{I} - (\mathbf{F} + \mathbf{GC})]^{-1}\mathbf{H} \quad (9-36)$$

This equation is satisfied if we choose \mathbf{H} equal to \mathbf{B} , and

$$\mathbf{A} = \mathbf{F} + \mathbf{GC} \quad (9-37)$$

Therefore, from (9-29) and (9-37), we write the observer state equations as

$$\mathbf{q}(k+1) = (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}\mathbf{y}(k) + \mathbf{B}\mathbf{u}(k) \quad (9-38)$$

and the design criterion (9-33) is satisfied. Note that \mathbf{G} is unspecified. Hence the observer is a dynamic system described by (9-38), with $\mathbf{y}(k)$ and $\mathbf{u}(k)$ as inputs and with the characteristic equation

$$|z\mathbf{I} - \mathbf{A} + \mathbf{GC}| = 0 \quad (9-39)$$

We call the observer of (9-38) a *prediction observer*, since the estimate at time $(k+1)T$ is based on the measurements at time kT .

Errors in Estimation

We now consider the errors in the state-estimation process. Define the error vector $\mathbf{e}(k)$ as

$$\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k) \quad (9-40)$$

Then, from (9-40), (9-26), and (9-38),

$$\begin{aligned} \mathbf{e}(k+1) &= \mathbf{x}(k+1) - \mathbf{q}(k+1) \\ &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) - (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) - \mathbf{GC}\mathbf{x}(k) - \mathbf{B}\mathbf{u}(k) \end{aligned} \quad (9-41)$$

or

$$\mathbf{e}(k+1) = (\mathbf{A} - \mathbf{GC})[\mathbf{x}(k) - \mathbf{q}(k)] = (\mathbf{A} - \mathbf{GC})\mathbf{e}(k) \quad (9-42)$$

Hence the error dynamics have a characteristic equation given by

$$|z\mathbf{I} - (\mathbf{A} - \mathbf{GC})| = 0 \quad (9-43)$$

which is the same as that of the observer [see (9-38)].

We look next at the sources of the errors in state estimation. First, we have assumed that we have an exact model of the physical system involved. For example, in (9-41) we assumed that the \mathbf{A} matrix in the plant's state equations is identical to that in the observer's equations. In a practical case we will have system models that only approximate the physical system. However, more effort expended in obtaining an accurate plant model will lead to improved state estimation.

A second source of errors in our state estimation is in the choice of initial

this error will tend to zero with increasing time (assuming exact models), with the dynamics given by (9-43).

The third source of errors is the plant disturbances and the sensor errors. The complete state equations of the plant, (9-26), become

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_1\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \end{aligned} \quad (9-44)$$

when the plant disturbance vector $\mathbf{w}(k)$ and the measurement inaccuracies $\mathbf{v}(k)$ are considered. Hence, if (9-44) is used in deriving the error state equations (9-42), we obtain

$$\mathbf{e}(k+1) = (\mathbf{A} - \mathbf{G}\mathbf{C})\mathbf{e}(k) + \mathbf{B}_1\mathbf{w}(k) - \mathbf{G}\mathbf{v}(k) \quad (9-45)$$

For this case, the errors in the state estimation will tend to zero if the estimator is stable (we will certainly require this); however, the excitation terms in (9-45) will prevent the errors from reaching a value of zero.

Error Dynamics

Consider now the observer state equations:

$$[\text{eq. (9-38)}] \quad \mathbf{q}(k+1) = (\mathbf{A} - \mathbf{G}\mathbf{C})\mathbf{q}(k) + \mathbf{G}\mathbf{y}(k) + \mathbf{B}\mathbf{u}(k)$$

All matrices in this equation are determined by the plant equations (9-26) except \mathbf{G} ; however, the observer design criterion (9-33) is satisfied independent of the choice of \mathbf{G} . Note that \mathbf{G} determines the error dynamics in (9-43). We generally use this fact in choosing \mathbf{G} . A characteristic polynomial $\alpha_e(z)$ is chosen for the error dynamics. Note from (9-38) and (9-43) that this is also the characteristic polynomial of the observer. Then

$$\alpha_e(z) = |z\mathbf{I} - (\mathbf{A} - \mathbf{G}\mathbf{C})| = z^n + \alpha_{n-1}z^{n-1} + \cdots + \alpha_1z + \alpha_0 \quad (9-46)$$

For the case of a single-output system [i.e., $\mathbf{y}(k)$ is a scalar],

$$\mathbf{G} = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{bmatrix} \quad (9-47)$$

and (9-46) yields n equations by equating coefficients. Hence the solution of these equations will yield the \mathbf{G} matrix of (9-47).

For this case Ackermann's equation may also be employed. First we compare (9-46) to (9-15).

$$[\text{eq. (9-15)}] \quad \alpha_c(z) = |z\mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{K})|$$

Ackermann's equation for the solution of this equation is given in (9-25).

$$[\text{eq. (9-25)}] \quad \mathbf{K} = [0 \ 0 \ \cdots \ 0 \ 1][\mathbf{B} \ \mathbf{A}\mathbf{B} \ \cdots \ \mathbf{A}^{n-2}\mathbf{B} \ \mathbf{A}^{n-1}\mathbf{B}]^{-1}\alpha_c(\mathbf{A})$$

Thus Ackermann's equation for \mathbf{G} in (9-46) is seen to be

$$\mathbf{G} = \alpha_e(\mathbf{A}) \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (9-48)$$

(The derivation of this result is given as Problem 9-28.)

Thus we see that we can design the observer, once we decide on an appropriate characteristic equation in (9-46). An approach usually suggested for choosing $\alpha_e(z)$ is to make the observer two to four times faster than the closed-loop control system. The fastest time constant of the closed-loop system, determined from the system characteristic equation,

$$|z\mathbf{I} - (\mathbf{A} - \mathbf{BK})| = 0 \quad (9-49)$$

is calculated first. The time constants of the observer are then set at a value equal to from one-fourth to one-half this fastest time constant.

The choice of \mathbf{G} can be approached from a different viewpoint. The observer state equation, from (9-38), can be written as

$$\mathbf{q}(k+1) = \mathbf{A}\mathbf{q}(k) + \mathbf{G}[\mathbf{y}(k) - \mathbf{C}\mathbf{q}(k)] + \mathbf{B}\mathbf{u}(k)$$

Thus the plant-observer system can be modeled as in Figure 9-5.

The choice of \mathbf{G} can be viewed in the following manner. In Figure 9-5, if $\mathbf{x}(k)$ and $\mathbf{q}(k)$ are approximately equal, there is little effect from the feedback through \mathbf{G} . Hence $\mathbf{q}(k)$ is determined principally from $\mathbf{u}(k)$. However, if the effects of disturbances on the plant cause $\mathbf{q}(k)$ to differ significantly from $\mathbf{x}(k)$, the effect of \mathbf{G} is much more important. Here the measurement of $\mathbf{y}(k)$ is more important in determining $\mathbf{q}(k)$ than in the first case. Thus we can view the choice of \mathbf{G} as relating

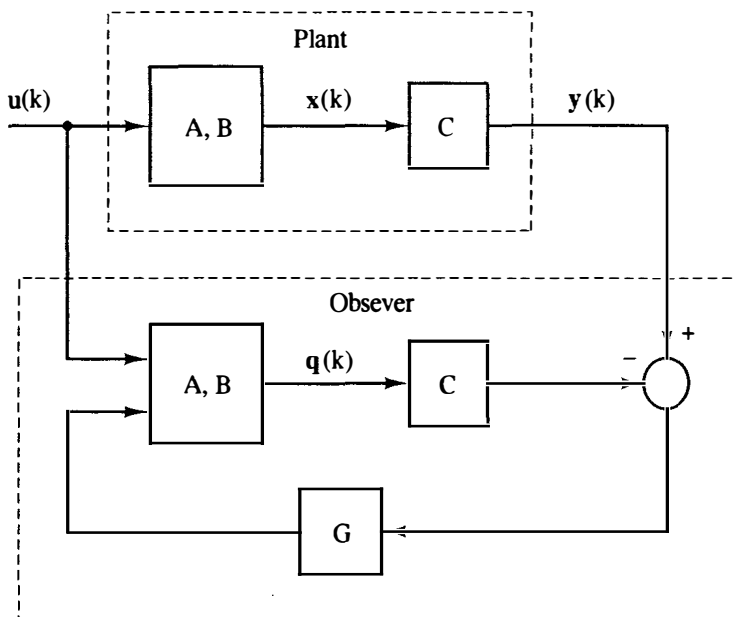


Figure 9-5 Different view of state estimation.

to the relative importance that we attach to the effects of $u(k)$ and to the effects of the disturbances on $x(k)$. And, of course, we will have the effects of measurement noise (inaccuracies) on $y(k)$. If these effects are great, the observer design cannot allow $y(k)$ to have a large weight in determining $q(k)$.

We will now summarize the paragraph above. We can view G as furnishing a correction input to the plant model in the observer in Figure 9-5, to account for unknowns in the plant. If the unknowns are significant, G should be relatively large. However, if measurement noise in $y(k)$ is significant, we cannot rely heavily on the measurement $y(k)$, and G should be relatively small. Hence, for practical systems, perhaps the best method for choosing G is through the use of an accurate simulation of the system of Figure 9-5 that includes both the disturbances and the measurement noise. Simulations should be run for different choices of G [obtained from different choices of $\alpha_e(z)$], with the final choice of G resulting from the best system response.

An example of the design of an observer will now be given.

Example 9.3



We will design an observer for the system of Example 9.1, which is the plant state equations

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} x(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] x(k) \end{aligned}$$

With the gain matrix

$$K = [4.52 \quad 1.12]$$

the closed-loop system characteristic equation is given by

$$\alpha_c(z) = z^2 - 1.776z + 0.819 = 0$$

The time constant of the roots of this equation, from Example 9.1, is 1.0 s. Thus we will choose the time constant of the observer to be 0.5 s. We choose the observer to be critically damped, with the roots

$$z = e^{-T/\tau} = e^{-0.1/0.5} = 0.819$$

The observer characteristic equation is then

$$\alpha_e(z) = (z - 0.819)^2 = z^2 - 1.638z + 0.671 = 0$$

The matrix G is given by (9-48):

$$G = \alpha_e(A) \begin{bmatrix} C \\ CA \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Now

$$\begin{aligned} \alpha_e(A) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix}^2 - 1.638 \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} + 0.671 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.033 & 0.0254 \\ 0 & 0.00763 \end{bmatrix} \\ \begin{bmatrix} C \\ CA \end{bmatrix}^{-1} &= \begin{bmatrix} 1 & 0 \\ 1 & 0.0952 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -10.51 & 10.51 \end{bmatrix} \end{aligned}$$

Then

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} 0.033 & 0.0254 \\ 0 & 0.00763 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -10.51 & 10.51 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.267 \\ 0.0802 \end{bmatrix} \end{aligned}$$

The system matrix of the estimator is given by

$$\mathbf{F} = \mathbf{A} - \mathbf{GC} = \begin{bmatrix} 0.733 & 0.0952 \\ -0.0802 & 0.905 \end{bmatrix}$$

From (9-38), the estimator's state equations are

$$\mathbf{q}(k+1) = \begin{bmatrix} 0.733 & 0.0952 \\ -0.0802 & 0.905 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.267 \\ 0.0802 \end{bmatrix} y(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k)$$

But

$$u(k) = -\mathbf{K}\mathbf{q}(k) = -4.52q_1(k) - 1.12q_2(k)$$

Then the estimator's state equations become

$$\mathbf{q}(k+1) = \begin{bmatrix} 0.711 & 0.0898 \\ -0.510 & 0.798 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.267 \\ 0.0802 \end{bmatrix} y(k)$$

Initial-condition responses for the observer-based control system are given in Figure 9-6. Two responses are given: one with

$$\mathbf{q}(0) = \mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and the other with

$$\mathbf{q}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Thus we can see an effect of not knowing the initial state of the system. It should be

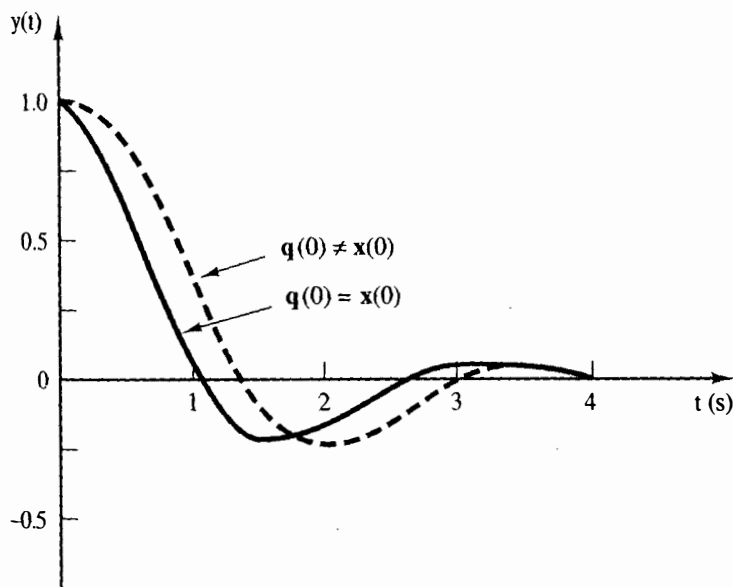


Figure 9-6 Initial conditions responses for Example 9.3.

noted that with $q(0)$ and $x(0)$ equal, the response of the observer-based control system is identical to that of the full-state feedback system of Example 9.1.

The effects of a certain disturbance, a unit-step function, will now be illustrated. The disturbance enters the system as given in (9-44), with, for example,

$$B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Figure 9-7 shows the disturbance response of both the observer-based system and the full-state feedback system with the unit-step disturbance. Since the observer does not take disturbances into account, the response of the observer-based control system differs significantly from that of the full-state feedback system. A MATLAB program that calculates the observer gains is given by

```
order = 2;
A = [1 .0952; 0 0.905];
B = [.00484; .0952];
C = [1 0];
alphae = [1 -1.638 0.671];
AMC=C;
AMCT=AMC;
for n=2:order
    AMCT=AMCT*A;
    for nn=1:order
        AMC(n,nn)=AMCT(1,nn);
    end
end
AMCI=inv(AMC);
CC=polyvalm(alphae,A);
CCC = CC*AMCI;
G=CCC(:,order)
```

Controller Transfer Function

We will now relate pole-assignment-estimator design to controller design by the procedures of Chapter 8. The observer equations are, from (9-38),

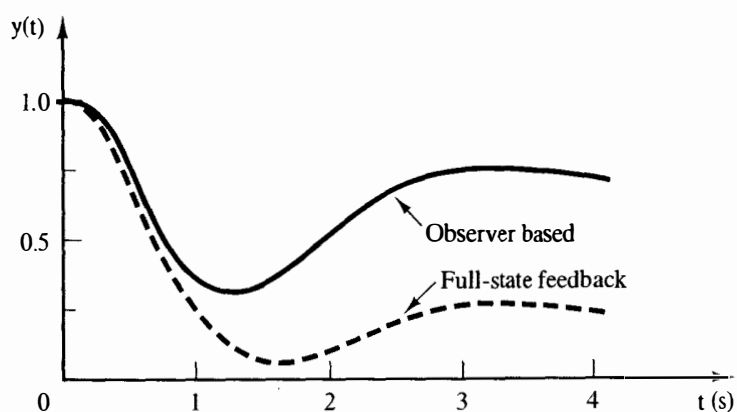


Figure 9-7 Disturbance responses for Example 9.3.

[eq. (9-38)] $\mathbf{q}(k + 1) = (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}y(k) + \mathbf{B}u(k)$

and the control law is given by

$$u(k) = -\mathbf{K}\mathbf{q}(k) \quad (9-50)$$

Substitution of the equation for $u(k)$ into the observer equations yields

$$\mathbf{q}(k + 1) = (\mathbf{A} - \mathbf{GC} - \mathbf{BK})\mathbf{q}(k) + \mathbf{G}y(k)$$

Taking the z -transform of this equation and solving for $\mathbf{Q}(z)$ yields

$$\mathbf{Q}(z) = (z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1} \mathbf{G}Y(z)$$

Substituting this equation into that for $U(z)$ results in the relationship

$$U(z) = -\mathbf{K}(z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1} \mathbf{G}Y(z)$$

Hence we can consider the control-observer combination to be a digital controller with the transfer function

$$D_{ce}(z) = -\frac{U(z)}{Y(z)} = \mathbf{K}(z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1} \mathbf{G} \quad (9-51)$$

and model the closed-loop system as shown in Figure 9-8a. Note that, for this controller, $-Y(z)$ is the *input*, and $U(z)$ is the *output*. In Figure 9-8a we show the control system as an equivalent unity-gain feedback system, with the system input equal to zero. Figure 9-8b gives the hardware configuration. We can also view the system with no input shown, as in Figure 9-8c. The system characteristic equation can be expressed as

$$1 + D_{ce}(z)G(z) = 0 \quad (9-52)$$

We will now determine the equivalent digital controller transfer function $D_{ce}(z)$ for the design of Example 9.3.

Example 9.4

The system equations for the system of Example 9.3 are

$$\begin{aligned} \mathbf{x}(k + 1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

with control gains

$$\mathbf{K} = [4.52 \quad 1.12]$$

In Example 9.3 the observer equations were calculated to be

$$\begin{aligned} \mathbf{q}(k + 1) &= (\mathbf{A} - \mathbf{BK} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}y(k) \\ &= \begin{bmatrix} 0.711 & 0.0898 \\ -0.510 & 0.798 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.267 \\ 0.0802 \end{bmatrix} y(k) \\ u(k) &= -\mathbf{K}\mathbf{q}(k) = -[4.52 \quad 1.12]\mathbf{q}(k) \end{aligned}$$

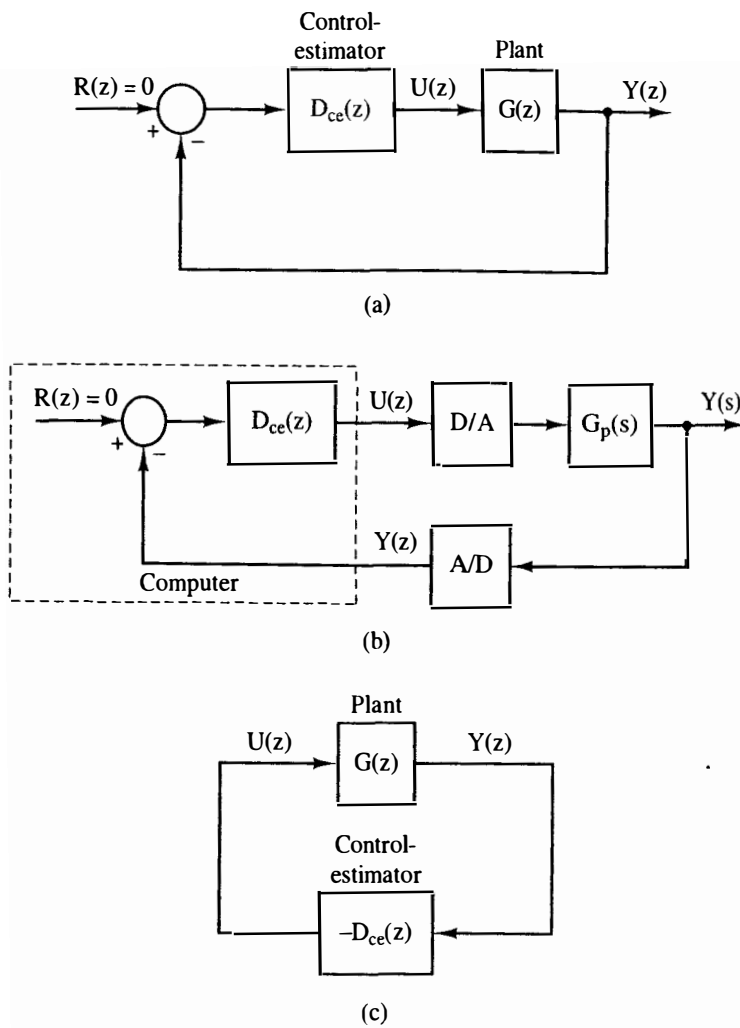


Figure 9-8 Digital-controller realization of the system.

Now, in (9-51),

$$(z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1} = \begin{bmatrix} z - 0.711 & -0.0898 \\ 0.510 & z - 0.798 \end{bmatrix}^{-1}$$

and

$$\Delta = |z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC}| = z^2 - 1.509z + 0.613$$

The digital controller transfer function is, from (9-51),

$$\begin{aligned} D_{ce}(z) &= \mathbf{K}(z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1}\mathbf{G} \\ &= [4.52 \quad 1.12] \frac{1}{\Delta} \begin{bmatrix} z - 0.798 & 0.0898 \\ -0.510 & z - 0.711 \end{bmatrix} \begin{bmatrix} 0.267 \\ 0.0802 \end{bmatrix} \\ &= \frac{1}{\Delta} [4.52z - 4.18 \quad 1.12z - 0.390] \begin{bmatrix} 0.267 \\ 0.0802 \end{bmatrix} \\ &= \frac{1.30z - 1.15}{z^2 - 1.509z + 0.613} \end{aligned}$$

It is seen that a second-order digital controller has been designed for this system by using a modern control approach.

Closed-Loop Characteristic Equation

As a final point, we must investigate the effects on the closed-loop system characteristic equation of the addition of the observer. For full-state feedback, the characteristic equation is given by

$$\alpha_c(z) = |z\mathbf{I} - \mathbf{A} + \mathbf{BK}| = 0$$

We will now derive the system characteristic equation for the observed-based control system.

To derive this equation, we will use the error variables of (9-40).

$$[\text{eq. (9-40)}] \quad \mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k)$$

The plant state equations of (9-26) can be expressed as

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) - \mathbf{BK}\mathbf{q}(k) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(k) + \mathbf{BK}\mathbf{e}(k) \quad (9-53)$$

The state equations for the error variables are

$$[\text{eq. (9-42)}] \quad \mathbf{e}(k+1) = (\mathbf{A} - \mathbf{GC})\mathbf{e}(k)$$

We can adjoin the variables of (9-53) and (9-42) into a single state vector, with the resulting equations

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{e}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{BK} & \mathbf{BK} \\ \mathbf{0} & \mathbf{A} - \mathbf{GC} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{e}(k) \end{bmatrix} \quad (9-54)$$

with the states of the observer a linear combination of $\mathbf{x}(k)$ and $\mathbf{e}(k)$, given in (9-40).

$$[\text{eq. (9-40)}] \quad \mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k)$$

Thus the characteristic equation of the state equations of (9-54) is also the closed-loop system characteristic equation. This equation is seen to be

$$|z\mathbf{I} - \mathbf{A} + \mathbf{BK}| |z\mathbf{I} - \mathbf{A} + \mathbf{GC}| = \alpha_c(z)\alpha_e(z) = 0 \quad (9-55)$$

We see then that the roots of the characteristic equation of the closed-loop system are the roots obtained by the pole-placement design plus those of the observer. Hence the pole-placement design is independent of the observer design.

Example 9.5

We will now calculate the characteristic equation for the system of Example 9.4. Now

$$G(z) = \frac{z-1}{z} \mathcal{Z} \left[\frac{1}{s^2(s+1)} \right] = \frac{0.00484z + 0.00468}{z^2 - 1.905z + 0.905}$$

Hence, from Example 9.4 and Figure 9-8, the closed-loop system characteristic equation can be expressed as

$$1 + D_{ce}(z)G(z) = 1 + \frac{1.30z - 1.15}{z^2 - 1.509z + 0.613} \left(\frac{0.00484z + 0.00468}{z^2 - 1.905z + 0.905} \right) = 0$$

This equation can be evaluated as

$$z^4 - 3.40z^3 + 4.40z^2 - 2.53z + 0.55 = 0$$

The product $\alpha_c(z)\alpha_e(z)$ is, from Examples 9.1 and 9.3,

$$\alpha_c(z)\alpha_e(z) = (z^2 - 1.776z + 0.819)(z^2 - 1.638z + 0.671)$$

and expansion of this equation yields the fourth-order polynomial above.

Closed-Loop State Equations

The transfer-function model of the closed-loop system is given in Figure 9-8. A state model of the closed-loop system will now be derived. From (9-26) and (9-50),

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) - \mathbf{B}\mathbf{K}\mathbf{q}(k)$$

and from (9-26), (9-38), and (9-50),

$$\mathbf{q}(k+1) = \mathbf{G}\mathbf{C}\mathbf{x}(k) + (\mathbf{A} - \mathbf{G}\mathbf{C} - \mathbf{B}\mathbf{K})\mathbf{q}(k)$$

We adjoin the foregoing two equations to form the closed-loop state model.

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{q}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{K} \\ \mathbf{G}\mathbf{C} & \mathbf{A} - \mathbf{G}\mathbf{C} - \mathbf{B}\mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{q}(k) \end{bmatrix} \quad (9-56)$$

This model is useful in writing simulations of the closed-loop system. If the disturbance inputs and the sensor-noise inputs are added as in (9-44), the closed-loop model is more complex (see Problem 9-29).

Example 9.6

The closed-loop state matrix of (9-56) will now be calculated for the system of Example 9.5. Now, from Example 9.3,

$$\mathbf{B}\mathbf{K} = \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} [4.52 \quad 1.12] = \begin{bmatrix} 0.0219 & 0.00542 \\ 0.430 & 0.107 \end{bmatrix}$$

$$\mathbf{G}\mathbf{C} = \begin{bmatrix} 0.267 \\ 0.0802 \end{bmatrix} [1 \quad 0] = \begin{bmatrix} 0.267 & 0 \\ 0.0802 & 0 \end{bmatrix}$$

From Example 9.4,

$$\mathbf{A} - \mathbf{G}\mathbf{C} - \mathbf{B}\mathbf{K} = \begin{bmatrix} 0.711 & 0.0898 \\ -0.510 & 0.798 \end{bmatrix}$$

The closed-loop system matrix, from (9-56), is then

$$\begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{K} \\ \mathbf{G}\mathbf{C} & \mathbf{A} - \mathbf{G}\mathbf{C} - \mathbf{B}\mathbf{K} \end{bmatrix} = \begin{bmatrix} 1 & 0.0952 & -0.0219 & -0.00542 \\ 0 & 0.905 & -0.430 & -0.107 \\ 0.267 & 0 & 0.711 & 0.0898 \\ 0.0802 & 0 & -0.510 & 0.798 \end{bmatrix}$$

Of course, the characteristic equation of this matrix is the same as that of (9-52) and (9-55). The interested reader may verify that the closed-loop system matrix calculated in this example has the same characteristic equation as (9-52), which was calculated in Example 9.5. Computer programs described in Appendix VI perform all of the calculations of this and all other examples of this chapter. It is obvious why these programs are a necessity.

This section has presented a brief introduction to observers. References 2, 4, 5, and 6 expand on the developments of this section.

9.4 REDUCED-ORDER OBSERVERS

In the example in Section 9.3, we estimated $x_1(k)$ [position] and $x_2(k)$ [velocity], given the measurement of position. However, if an accurate measurement of position is available, it is not reasonable to try to estimate position; we already know it. In general, if accurate measurements of certain states are made, one needs to estimate only the remaining states, and the accurately measured signals are then used directly for feedback. The resulting observer is called a reduced-order observer. However, if the measurements are relatively inaccurate (noisy), the full-order observer may yield better results.

To develop the design equations for the reduced-order observer, we first partition the state vector as

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_a(k) \\ \mathbf{x}_b(k) \end{bmatrix}$$

where $\mathbf{x}_a(k)$ are the states to be measured and $\mathbf{x}_b(k)$ are the states to be estimated. Then the plant state equation of (9-26) can be partitioned as

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_a(k+1) \\ \mathbf{x}_b(k+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_{aa} & \mathbf{A}_{ab} \\ \mathbf{A}_{ba} & \mathbf{A}_{bb} \end{bmatrix} \begin{bmatrix} \mathbf{x}_a(k) \\ \mathbf{x}_b(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_a \\ \mathbf{B}_b \end{bmatrix} \mathbf{u}(k) \\ \mathbf{y}(k) &= [\mathbf{I} \quad 0] \begin{bmatrix} \mathbf{x}_a(k) \\ \mathbf{x}_b(k) \end{bmatrix} \end{aligned} \quad (9-57)$$

Note that in this case we are considering both multiple inputs and multiple outputs. The equations for the measured states can be written as

$$\mathbf{x}_a(k+1) = \mathbf{A}_{aa} \mathbf{x}_a(k) + \mathbf{A}_{ab} \mathbf{x}_b(k) + \mathbf{B}_a \mathbf{u}(k)$$

Collecting all the known terms on the left side of the equation, we write

$$\mathbf{x}_a(k+1) - \mathbf{A}_{aa} \mathbf{x}_a(k) - \mathbf{B}_a \mathbf{u}(k) = \mathbf{A}_{ab} \mathbf{x}_b(k) \quad (9-58)$$

For the reduced-order observer we consider the left side to be the "known measurements." From (9-57), the equations for the estimated states are

$$\mathbf{x}_b(k+1) = \mathbf{A}_{ba} \mathbf{x}_a(k) + \mathbf{A}_{bb} \mathbf{x}_b(k) + \mathbf{B}_b \mathbf{u}(k) \quad (9-59)$$

The term $[\mathbf{A}_{ba} \mathbf{x}_a(k) + \mathbf{B}_b \mathbf{u}(k)]$ is then considered to be the "known inputs." We now compare the state equations for the full-order observer to those for the reduced-order observer.

$$[\text{eq. (9-26)}] \quad \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

$$[\text{eq. (9-59)}] \quad \mathbf{x}_b(k+1) = \mathbf{A}_{bb} \mathbf{x}_b(k) + [\mathbf{A}_{ba} \mathbf{x}_a(k) + \mathbf{B}_b \mathbf{u}(k)]$$

and

$$[\text{eq. (9-26)}] \quad \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k)$$

$$[\text{eq. (9-58)}] \quad \mathbf{x}_a(k+1) - \mathbf{A}_{aa}\mathbf{x}_a(k) - \mathbf{B}_a\mathbf{u}(k) = \mathbf{A}_{ab}\mathbf{x}_b(k)$$

We then obtain the reduced-order observer equations by making the following substitutions into the full-order observer equations (9-38):

$$\mathbf{x}(k) \leftarrow \mathbf{x}_b(k)$$

$$\mathbf{A} \leftarrow \mathbf{A}_{bb}$$

$$\mathbf{B}\mathbf{u}(k) \leftarrow \mathbf{A}_{ba}\mathbf{x}_a(k) + \mathbf{B}_b\mathbf{u}(k)$$

$$\mathbf{y}(k) \leftarrow \mathbf{x}_a(k+1) - \mathbf{A}_{aa}\mathbf{x}_a(k) - \mathbf{B}_a\mathbf{u}(k)$$

$$\mathbf{C} \leftarrow \mathbf{A}_{ab}$$

If we make these substitutions into (9-38),

$$[\text{eq. (9-38)}] \quad \mathbf{q}(k+1) = (\mathbf{A} - \mathbf{G}\mathbf{C})\mathbf{q}(k) + \mathbf{G}\mathbf{y}(k) + \mathbf{B}\mathbf{u}(k)$$

we obtain the equations

$$\begin{aligned} \mathbf{q}_b(k+1) &= (\mathbf{A}_{bb} - \mathbf{G}\mathbf{A}_{ab})\mathbf{q}_b(k) + \mathbf{G}[\mathbf{x}_a(k+1) - \mathbf{A}_{aa}\mathbf{x}_a(k) - \mathbf{B}_a\mathbf{u}(k)] \\ &\quad + \mathbf{A}_{ba}\mathbf{x}_a(k) + \mathbf{B}_b\mathbf{u}(k) \end{aligned} \quad (9-60)$$

From (9-57),

$$\mathbf{y}(k) = \mathbf{x}_a(k) \quad (9-61)$$

Then (9-60) can be written as

$$\begin{aligned} \mathbf{q}_b(k+1) &= (\mathbf{A}_{bb} - \mathbf{G}\mathbf{A}_{ab})\mathbf{q}_b(k) + \mathbf{G}\mathbf{y}(k+1) + (\mathbf{A}_{ba} - \mathbf{G}\mathbf{A}_{aa})\mathbf{y}(k) \\ &\quad + (\mathbf{B}_b - \mathbf{G}\mathbf{B}_a)\mathbf{u}(k) \end{aligned} \quad (9-62)$$

The following points should be made about the reduced-order observer. The observer characteristic equation is

$$\alpha_e(z) = |z\mathbf{I} - \mathbf{A}_{bb} + \mathbf{G}\mathbf{A}_{ab}| = 0 \quad (9-63)$$

and \mathbf{G} is determined from (9-63) in the same manner that the \mathbf{G} matrix is obtained from (9-46) for the full-order observer. For the case of a single measurement [i.e., $\mathbf{y}(k)$ is $x_1(k)$] Ackermann's formula is given by

$$\mathbf{G} = \alpha_e(\mathbf{A}_{bb}) \begin{bmatrix} \mathbf{A}_{ab} \\ \mathbf{A}_{ab}\mathbf{A}_{bb} \\ \vdots \\ \mathbf{A}_{ab}\mathbf{A}_{bb}^{n-2} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (9-64)$$

Note also that, in (9-62), the measurements $\mathbf{y}(k+1)$ is required to estimate $\mathbf{q}_b(k+1)$. However, for the full-order observer, in (9-38), $\mathbf{q}(k+1)$ is estimated using only the measurements $\mathbf{y}(k)$.

As in the case of the full-order prediction observer, we can derive the transfer function $D_{ce}(z)$, in Figure 9-8, of the equivalent digital controller. Since the derivation of the transfer function follows the same procedure as in the full-order observer case, this derivation is given as Problem 9-16. It is assumed that $y(k) = x_1(k)$, and that the gain matrix \mathbf{K} is partitioned as

$$\begin{aligned} u(k) &= -\mathbf{K}[y(k) \quad \mathbf{q}_b(k)]^T = -[K_1 \quad \mathbf{K}_b][y(k) \quad \mathbf{q}_b(k)]^T \\ &= -K_1 y(k) - \mathbf{K}_b \mathbf{q}_b(k) \end{aligned} \quad (9-65)$$

where $[\cdot]^T$ indicates the transpose of the matrix $[\cdot]$. The derivation results in the transfer function

$$\begin{aligned} D_{ce}(z) &= \frac{-U(z)}{Y(z)} \\ &= K_1 + \mathbf{K}_b[z\mathbf{I} - \mathbf{A}_{bb} + \mathbf{G}\mathbf{A}_{ab} + (\mathbf{B}_b - \mathbf{G}\mathbf{B}_a)\mathbf{K}_b]^{-1} \\ &\quad \cdot [\mathbf{G}z + \{\mathbf{A}_{ba} - \mathbf{G}\mathbf{A}_{aa} - K_1(\mathbf{B}_b - \mathbf{G}\mathbf{B}_a)\}] \end{aligned} \quad (9-66)$$

An example of the reduced-order estimator will now be given.

Example 9.7



We will again consider the design of the system of Example 9.1. The plant model is given by

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

Thus we are measuring position, $x_1(k)$, and will estimate velocity, $x_2(k)$. The closed-loop system characteristic equation was chosen to be

$$\alpha_c(z) = z^2 - 1.776z + 0.819 = 0$$

In Example 9.3 we chose the estimator characteristic-equation roots to be at $z = 0.819$; we will make the same choice here. However, the reduced-order observer is first order; hence

$$\alpha_e(z) = z - 0.819 = 0$$

From the plant state equations above, and (9-57), the partitioned matrices are seen to be

$$\begin{aligned} \mathbf{A}_{aa} &= 1 & \mathbf{A}_{ab} &= 0.0952 & \mathbf{B}_a &= 0.00484 \\ \mathbf{A}_{ba} &= 0 & \mathbf{A}_{bb} &= 0.905 & \mathbf{B}_b &= 0.0952 \end{aligned}$$

Thus we have all the terms required for Ackermann's formula in (9-64).

$$\begin{aligned} \mathbf{G} &= \alpha_e(\mathbf{A}_{bb})[\mathbf{A}_{ab}]^{-1}[1] = [0.905 - 0.819][0.0952]^{-1}[1] \\ &= 0.903 \end{aligned}$$

The observer equation is given by (9-62):

$$\begin{aligned} q(k+1) &= [0.905 - (0.903)(0.0952)]q(k) + 0.903y(k+1) + \\ &\quad [0 - (0.903)(1)]y(k) + [0.0952 - (0.903)(0.00484)]u(k) \end{aligned}$$

or

$$q(k+1) = 0.819q(k) + 0.903y(k+1) - 0.903y(k) + 0.0908u(k)$$

Here $q(k)$ is the estimate of velocity, $x_2(k)$. Since we have considered the measurement $y(k)$ to be the measurement at the present time, the implementation of the observer is more obvious if we replace k with $(k+1)$ in the observer equation:

$$q(k) = 0.819q(k-1) + 0.903y(k) - 0.903y(k-1) + 0.0908u(k-1)$$

and $q(k)$ is the estimate at the present time. From Example 9.1 the control law is given by

$$u(k) = -4.52x_1(k) - 1.12x_2(k)$$

which is implemented as

$$u(k) = -4.52y(k) - 1.12q(k)$$

Hence we can write the observer equation as

$$\begin{aligned} q(k+1) &= 0.819q(k) + 0.903y(k+1) - 0.903y(k) \\ &\quad + 0.0908[-4.52y(k) - 1.12q(k)] \end{aligned}$$

or

$$q(k+1) = 0.717q(k) + 0.903y(k+1) - 1.313y(k)$$

The control system is then implemented as follows. A measurement $y(k)$ is made at $t = kT$. The observer state is calculated from

$$q(k) = 0.717q(k-1) + 0.903y(k) - 1.313y(k-1)$$

Then the control input is calculated, using

$$u(k) = -4.52y(k) - 1.12q(k)$$

The initial-condition response, obtained by simulation, is approximately the same as that obtained using full-state feedback. In addition, the effects of the disturbance input are less than those for the full-order observer system.

The observer gain for this example is calculated by the MATLAB program

```
order = 2;
A = [1 .0952; 0 0.905];
B = [.00484; .0952];
C = [1 0];
alphae = [1 -0.819];
Aaa = A(1,1); Aab = A(1,2); Aba = A(2,1); Abb = A(2,2);
Ba = B(1,1); Bb = B(2,1);
AMC=Aab;
AMCI=inv(AMC);
CC=polyvalm(alphae,Abb);
G = CC*AMCI
```

Example 9.8

We will now calculate the transfer function of the controller-estimator for the system of Example 9.7. From Example 9.7, $G = 0.903$ and

$$\begin{array}{llll} A_{aa} = 1 & A_{ab} = 0.0952 & B_a = 0.00484 & K_1 = 4.52 \\ A_{ba} = 0 & A_{bb} = 0.905 & B_b = 0.0952 & K_b = 1.12 \end{array}$$

and from (9-66),

$$\begin{aligned} D_{ce}(z) &= \frac{-U(z)}{Y(z)} \\ &= K_1 + K_b[z - A_{bb} + GA_{ab} + (B_b - GB_a)K_b]^{-1} \\ &\quad \cdot [Gz + \{A_{ba} - GA_{aa} - K_1(B_b - GB_a)\}] \\ &= 4.52 + 1.12[z - 0.905 + (0.903)(0.0952) + \{0.0952 - (0.903)(0.00484)\}1.12]^{-1} \\ &\quad \cdot \{0.903z + [0 - (0.903)(1) - 4.52\{0.0952 - (0.903)(0.00484)\}]\} \\ &= 4.52 + \frac{1.12(0.903z - 1.314)}{z - 0.717} = \frac{5.53z - 4.71}{z - 0.717} \\ &= \frac{5.53(z - 0.852)}{z - 0.717} \end{aligned}$$

Since the zero is closer to $z = 1$ than is the pole, the control-estimator is phase lead. This must be true independent of the design procedure, since the system is designed to increase the speed of response, as specified in Example 9.1.

9.5 CURRENT OBSERVERS

The estimator developed in Section 9.3 is a prediction observer, since the estimate of $\mathbf{x}(k)$ is based on the measurement $\mathbf{y}(k - 1)$. However, the reduced-order observer of Section 9.4 estimates states at time kT using the measurement at time kT . We call this type of estimator a *current estimator*, or current observer. We will now consider a full-order current estimator.

As before, the system model is given by

$$\begin{aligned} [\text{eq. (9-26)}] \quad \mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned}$$

We wish to estimate the state vector $\mathbf{x}(k)$ with the vector $\mathbf{q}(k)$. One form of a full-order current observer is given by the two equations

$$\begin{aligned} \bar{\mathbf{q}}(k + 1) &= \mathbf{A}\mathbf{q}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{q}(k + 1) &= \bar{\mathbf{q}}(k + 1) + \mathbf{G}[\mathbf{y}(k + 1) - \mathbf{C}\bar{\mathbf{q}}(k + 1)] \end{aligned} \quad (9-67)$$

In these equations, $\bar{\mathbf{q}}(k + 1)$ is a first estimate of the state at time $(k + 1)T$, based on the dynamics of the system and on the signals at kT . This estimate is then

corrected in the second equation when the measurement at time $(k + 1)T$ arrives. The estimator gain is \mathbf{G} , and determines the weight placed on the difference between measurement at $(k + 1)T$ and what we expect that measurement to be at that time. The final estimate is $\mathbf{q}(k + 1)$.

In (9-67), the first estimate $\bar{\mathbf{q}}(k + 1)$ can be eliminated by substituting the first equation into the second one. The single equation for the estimate is then

$$\mathbf{q}(k + 1) = [\mathbf{A} - \mathbf{GCA}]\mathbf{q}(k) + [\mathbf{B} - \mathbf{GCB}]\mathbf{u}(k) + \mathbf{Gy}(k + 1) \quad (9-68)$$

We see then that the estimate at time $(k + 1)T$ is based on the measurement at $(k + 1)T$. The dynamics of this estimator are described by the characteristic equation

$$|z\mathbf{I} - \mathbf{A} + \mathbf{GCA}| = 0 \quad (9-69)$$

It can be shown that this estimator has the same property as the prediction estimator; the transfer matrix $\mathbf{Q}(z)/\mathbf{U}(z)$ is equal to the transfer matrix $\mathbf{X}(z)/\mathbf{U}(z)$. The proof of this property is straightforward, and is given as Problem 9-17.

As in the prediction-observer case, we define the estimation-error vector $\mathbf{e}(k)$ by the relationship

$$\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k) \quad (9-70)$$

Then

$$\begin{aligned} \mathbf{e}(k + 1) &= \mathbf{x}(k + 1) - \mathbf{q}(k + 1) \\ &= \mathbf{Ax}(k) + \mathbf{Bu}(k) - [\mathbf{A} - \mathbf{GCA}]\mathbf{q}(k) \\ &\quad - [\mathbf{B} - \mathbf{GCB}]\mathbf{u}(k) - \mathbf{GC}[\mathbf{Ax}(k) + \mathbf{Bu}(k)] \\ &= [\mathbf{A} - \mathbf{GCA}][\mathbf{x}(k) - \mathbf{q}(k)] = [\mathbf{A} - \mathbf{GCA}]\mathbf{e}(k) \end{aligned} \quad (9-71)$$

from (9-26) and (9-68). Hence the error vector has dynamics with the same characteristic equation as the estimator, (9-69).

The characteristic equation for the prediction observer is

$$[\text{eq. (9-43)}] \quad |z\mathbf{I} - \mathbf{A} + \mathbf{GC}| = 0$$

and for the current observer is

$$[\text{eq. (9-69)}] \quad |z\mathbf{I} - \mathbf{A} + \mathbf{GCA}| = 0$$

Thus, for a single-input single-output system, Ackerman's formula for the current observer is obtained from that for the prediction observer, (9-48), by replacing \mathbf{C} with \mathbf{CA} ; hence, for the current observer,

$$\mathbf{G} = \alpha_e(\mathbf{A}) \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^n \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (9-72)$$

A pole-placement design implemented with a current observer can be viewed as a digital-controller design as shown in Figure 9-8, with the controller-estimator

transfer function given by

$$D_{ce}(z) = zK[zI - A + GCA + BK - GCBK]^{-1}G \quad (9-73)$$

(see Problem 9-17). The current observer design will now be illustrated with an example.

Example 9.9

We will again consider the system of the earlier examples:

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

with control gains

$$\mathbf{K} = [4.52 \quad 1.12]$$

We will use the same observer characteristic equation as used for the prediction observer of Example 9.3.

$$\alpha_e(z) = z^2 - 1.638z + 0.671$$

From Example 9.3,

$$\alpha_e(\mathbf{A}) = \begin{bmatrix} 0.033 & 0.0254 \\ 0 & 0.00763 \end{bmatrix}$$

Also,

$$\mathbf{CA} = [1 \quad 0.0952]$$

and thus

$$\mathbf{CA}^2 = (\mathbf{CA})\mathbf{A} = [1 \quad 0.0952] \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} = [1 \quad 0.1814]$$

Then

$$\begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0.0952 \\ 1 & 0.1814 \end{bmatrix}^{-1} = \begin{bmatrix} 2.104 & -1.104 \\ -11.60 & 11.60 \end{bmatrix}$$

From (9-72),

$$\begin{aligned} \mathbf{G} &= \alpha_e(\mathbf{A}) \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.033 & 0.0254 \\ 0 & 0.00763 \end{bmatrix} \begin{bmatrix} 2.104 & -1.104 \\ -11.60 & 11.60 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.033 & 0.0254 \\ 0 & 0.00763 \end{bmatrix} \begin{bmatrix} -1.104 \\ 11.60 \end{bmatrix} = \begin{bmatrix} 0.258 \\ 0.0885 \end{bmatrix} \end{aligned}$$

The difference equation of the observer in (9-68) will now be evaluated.

$$\begin{aligned} \mathbf{GCA} &= \begin{bmatrix} 0.258 \\ 0.0885 \end{bmatrix} [1 \quad 0] \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \\ &= \begin{bmatrix} 0.258 \\ 0.0885 \end{bmatrix} [1 \quad 0.0952] \\ &= \begin{bmatrix} 0.258 & 0.0246 \\ 0.0885 & 0.00843 \end{bmatrix} \end{aligned}$$

and

$$\mathbf{A} - \mathbf{GCA} = \begin{bmatrix} 0.742 & 0.0706 \\ -0.0885 & 0.897 \end{bmatrix}$$

Also, evaluation of $(\mathbf{B} - \mathbf{GCB})$ yields

$$\mathbf{B} - \mathbf{GCB} = \begin{bmatrix} 0.00359 \\ 0.0948 \end{bmatrix}$$

We see then from (9-68) that the observer is implemented as

$$\mathbf{q}(k+1) = \begin{bmatrix} 0.742 & 0.0706 \\ -0.0885 & 0.897 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.00359 \\ 0.0948 \end{bmatrix} u(k) + \begin{bmatrix} 0.258 \\ 0.0885 \end{bmatrix} y(k+1)$$

The gain vector \mathbf{G} is calculated by the MATLAB program

```
order = 2;
A = [1 .0952; 0 0.905];
B = [.00484; .0952];
C = [1 0];
alphae = [1 -1.638 0.671];
AMC=C*A;
AMCT=AMC;
for n=2:order
    AMCT=AMCT*A;
    for nn=1:order
        AMC(n,nn)=AMCT(1,nn);
    end
end
AMCI=inv(AMC);
CC=polyvalm(alphae,A);
CCC = CC*AMCI;
G=CCC(:,order)
```

Example 9.10

This example is a continuation of the last example. The transfer function of the controller estimator will be calculated from (9-73). From Example 9.9,

$$\mathbf{A} - \mathbf{GCA} = \begin{bmatrix} 0.742 & 0.0706 \\ -0.0885 & 0.897 \end{bmatrix}, \quad \mathbf{B} - \mathbf{GCB} = \begin{bmatrix} 0.00359 \\ 0.0948 \end{bmatrix}$$

Hence

$$(\mathbf{B} - \mathbf{GCB})\mathbf{K} = \begin{bmatrix} 0.00359 \\ 0.0948 \end{bmatrix} \begin{bmatrix} 4.52 & 1.12 \end{bmatrix} = \begin{bmatrix} 0.0219 & 0.00542 \\ 0.4303 & 0.1066 \end{bmatrix}$$

and

$$\mathbf{A} - \mathbf{GCA} - \mathbf{BK} + \mathbf{GCBK} = \begin{bmatrix} 0.726 & 0.0666 \\ -0.517 & 0.791 \end{bmatrix}$$

$$\begin{aligned}
[z\mathbf{I} - \mathbf{A} + \mathbf{GCA} + \mathbf{BK} - \mathbf{GCBK}]^{-1} &= \begin{bmatrix} z - 0.726 & -0.0666 \\ 0.157 & z - 0.791 \end{bmatrix}^{-1} \\
&= \frac{1}{\Delta} \begin{bmatrix} z - 0.791 & 0.0666 \\ -0.517 & z - 0.726 \end{bmatrix}, \\
\Delta &= |z\mathbf{I} - \mathbf{A} + \mathbf{GCA} + \mathbf{BK} - \mathbf{GCBK}| \\
&= z^2 - 1.52z + 0.609
\end{aligned}$$

Then, from (9-73),

$$\begin{aligned}
D_{ce}(z) &= z\mathbf{K}[z\mathbf{I} - \mathbf{A} + \mathbf{GCA} + \mathbf{BK} - \mathbf{GCBK}]^{-1}\mathbf{G} \\
&= \frac{z}{\Delta} \begin{bmatrix} 4.52 & 1.12 \end{bmatrix} \begin{bmatrix} z - 0.791 & 0.0666 \\ -0.517 & z - 0.726 \end{bmatrix} \begin{bmatrix} 0.258 \\ 0.0885 \end{bmatrix} \\
&= \frac{z}{\Delta} \begin{bmatrix} 4.52 & 1.12 \end{bmatrix} \begin{bmatrix} 0.258z - 0.198 \\ 0.0885z - 0.1976 \end{bmatrix} \\
&= \frac{1.27z^2 - 1.12z}{z^2 - 1.52z + 0.609}
\end{aligned}$$

A problem can occur in the implementation of observer-based controllers. Thus far we have not considered the relative stability of these control systems. A system that has adequate stability margins is said to be *robust*; observer-based control systems may not be robust in terms of the gain and phase margins that appear at the input to the plant. To determine these stability margins, we see from Figure 9-8 that the frequency response for the open-loop function $D_{ce}(z)G(z)$ must be calculated. This was done for the prediction-observer system of Example 9.4 and the current-observer system of Example 9.9. In both systems the phase margin is 49° and the gain margins are greater than 8 dB. In these systems no relative-stability problems appear. This problem is discussed further, when Kalman filters (which are optimal current observers) are discussed in Chapter 10.

As in the case of the prediction observer in (9-56), the state model of the closed-loop system employing the current observer can be derived as

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{q}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{GCA} & \mathbf{A} - \mathbf{GCA} - \mathbf{BK} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{q}(k) \end{bmatrix} \quad (9-74)$$

The derivation of this equation is straightforward and is given as an exercise in Problem 9-30.

9.6 CONTROLLABILITY AND OBSERVABILITY

In the preceding sections Ackermann's formula was useful in both pole-assignment design, (9-25), and in observer design, (9-48). In pole-assignment design it was necessary that the inverse of the matrix

$$[\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}] \quad (9-75)$$

exist, and in prediction observer design it was necessary that the inverse of the matrix

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (9-76)$$

exist. We will now relate the existence of the inverses of (9-75) and (9-76) to the important concepts of controllability and observability, respectively.

We introduce the concept of *controllability* with respect to the system of Figure 9-9. The system characteristic equation is given by

$$(z - 0.9)(z - 0.8) = 0$$

However, the mode of the transient response $(0.9)^k$ is not excited by the input $u(k)$ and hence cannot be controlled by $u(k)$. This system is said to be *uncontrollable*.

Definition 1. A system

$$\mathbf{x}(k + 1) = \mathbf{Ax}(k) + \mathbf{Bu}(k) \quad (9-77)$$

is controllable provided that there exists a sequence of input $\mathbf{u}(0), \mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(N)$ that will translate the system from any initial state $\mathbf{x}(0)$ to any final state $\mathbf{x}(N)$, with N finite.

Note that in the system of Figure 9-9, the input $u(k)$ has no influence on the state of the upper block; thus the system is uncontrollable.

We will now derive the conditions for the system of (9-77) to be controllable.

Now

$$\mathbf{x}(1) = \mathbf{Ax}(0) + \mathbf{Bu}(0)$$

$$\mathbf{x}(2) = \mathbf{Ax}(1) + \mathbf{Bu}(1) \quad (9-78)$$

$$= \mathbf{A}^2 \mathbf{x}(0) + \mathbf{ABu}(0) + \mathbf{Bu}(1)$$

...

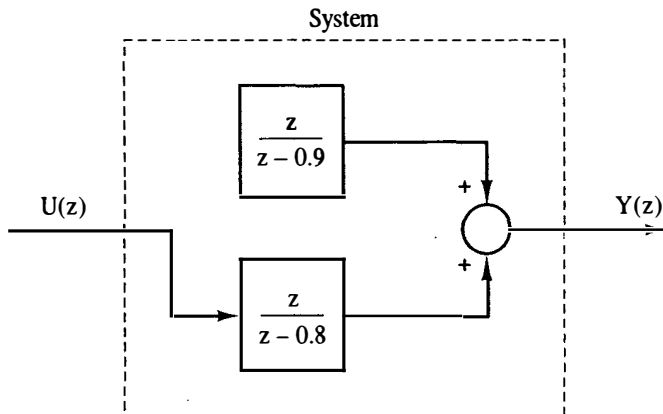


Figure 9-9 Uncontrollable system.

$$\begin{aligned}
 \mathbf{x}(N) &= \mathbf{A}^N \mathbf{x}(0) + \mathbf{A}^{N-1} \mathbf{B} \mathbf{u}(0) + \cdots + \mathbf{A} \mathbf{B} \mathbf{u}(N-2) + \mathbf{B} \mathbf{u}(N-1) \\
 &= \mathbf{A}^N \mathbf{x}(0) + [\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \cdots \quad \mathbf{A}^{N-1} \mathbf{B}] \begin{bmatrix} \mathbf{u}(N-1) \\ \mathbf{u}(N-2) \\ \vdots \\ \mathbf{u}(0) \end{bmatrix}
 \end{aligned} \tag{9-79}$$

Hence, with $\mathbf{x}(N)$ and $\mathbf{x}(0)$ known, this equation can be written as

$$[\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \cdots \quad \mathbf{A}^{N-1} \mathbf{B}] \begin{bmatrix} \mathbf{u}(N-1) \\ \mathbf{u}(N-2) \\ \vdots \\ \mathbf{u}(0) \end{bmatrix} = \mathbf{x}(N) - \mathbf{A}^N \mathbf{x}(0) \tag{9-80}$$

Since the order of the state vector $\mathbf{x}(k)$ is n , then (9-80) yields n linear simultaneous equations. Hence, for a solution to exist, the rank of the coefficient matrix [which is (9-75) for N equal to n]

$$[\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \cdots \quad \mathbf{A}^{N-1} \mathbf{B}] \tag{9-81}$$

must be n [7]. For the case of a single input, \mathbf{B} is a column matrix and (9-81) is $n \times n$ for N equal to n . Then the inverse of (9-81) must exist; this is also the condition for the existence of the solution to Ackermann's formula, in (9-25).

We consider next the concept of *observability*.

Definition 2. A system

$$\begin{aligned}
 \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\
 \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)
 \end{aligned} \tag{9-82}$$

is observable provided that the initial state $\mathbf{x}(0)$, for any $\mathbf{x}(0)$, can be calculated from the N measurements $\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(N-1)$, with N finite.

A system that is obviously unobservable is shown in Figure 9-10, since the state of the upper block does not contribute to the output $\mathbf{y}(k)$.

The derivation of the criteria for observability closely parallels that for control-

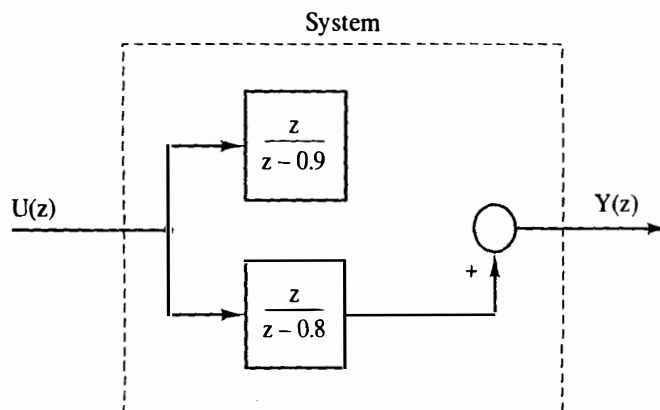


Figure 9-10 Unobservable system.

lability. To simplify the derivation, we will assume that $u(k)$ is zero; the derivation for $u(k)$ not equal to zero is similar, and is given as Problem 9-18. Now, from (9-82),

$$\begin{aligned} y(0) &= Cx(0) \\ y(1) &= Cx(1) = CAx(0) \\ &\vdots \\ y(N-1) &= Cx(N-1) = CA^{N-1}x(0) \end{aligned} \quad (9-83)$$

or

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} x(0) \quad (9-84)$$

Hence, by the same arguments as for controllability, the rank of the coefficient matrix [which is (9-76) for N equal to n]

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} \quad (9-85)$$

must be n . For the case of a single output, C is a row matrix and (9-85) is $n \times n$ for N equal to n . Then the inverse of (9-85) must exist; this is the condition for the existence of the solution to Ackermann's formula, (9-48).

Example 9.11

We will test the system of Figure 9-11 for both controllability and observability. This system is composed of two cascaded first-order subsystems, each of which is controllable and observable, which is seen by inspection. The system state equations are given by

$$\begin{aligned} x(k+1) &= \begin{bmatrix} -0.2 & 0 \\ -1 & 0.8 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [-1 \quad 1]x(k) + u(k) \end{aligned}$$

First we will test for controllability, using (9-81).

$$AB = \begin{bmatrix} -0.2 & 0 \\ -1 & 0.8 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.2 \\ -0.2 \end{bmatrix}$$

and thus

$$[B \quad AB] = \begin{bmatrix} 1 & -0.2 \\ 1 & -0.2 \end{bmatrix}$$

This matrix is of rank 1 and its inverse does not exist; hence the system is uncontrollable. Next we will test for observability, using (9-85).

$$CA = [-1 \quad 1] \begin{bmatrix} -0.2 & 0 \\ -1 & 0.8 \end{bmatrix} = [-0.8 \quad 0.8]$$

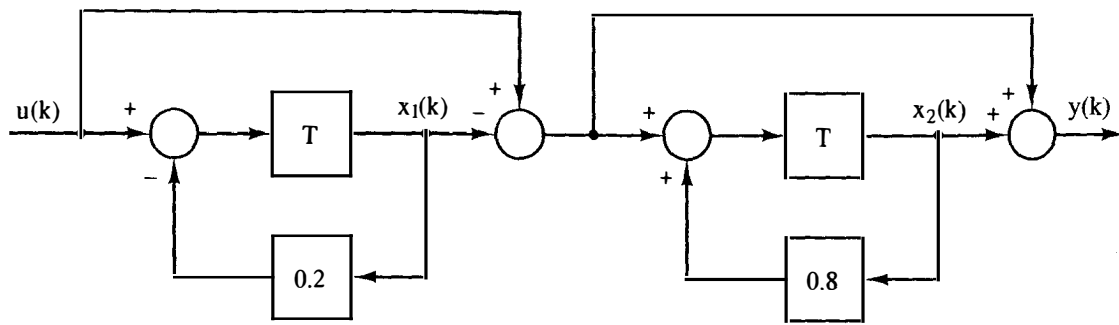


Figure 9-11 System for Example 9.11.

and thus

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -0.8 & 0.8 \end{bmatrix}$$

This matrix is also of rank 1; hence the system is unobservable. It is instructive to calculate some transfer functions. By Mason's gain formula and from Figure 9-11,

$$\frac{Y(z)}{U(z)} = \frac{1(1 - 0.6z^{-1} - 0.16z^{-2}) - z^{-1}(1 - 0.8z^{-1}) + z^{-1}(1 + 0.2z^{-1}) - z^{-2}}{1 + 0.2z^{-1} - 0.8z^{-1} - 0.16z^{-2}} = 1$$

$$\frac{X_2(z)}{U(z)} = \frac{z^{-1}(1 + 0.2z^{-1}) - z^{-2}}{1 - 0.6z^{-1} - 1.6z^{-2}} = \frac{z^{-1}}{1 + 0.2z^{-1}}$$

$$\frac{X_1(z)}{U(z)} = \frac{z^{-1}}{1 + 0.2z^{-1}}$$

$$\frac{Y(z)}{X_1(z)} = \frac{Y(z)}{U(z)} \frac{U(z)}{X_1(z)} = \frac{1 + 0.2z^{-1}}{z^{-1}}$$

From the transfer function $X_2(z)/U(z)$, we see that the mode $(0.8)^k$ is not excited by the input. From the transfer function $Y(z)/X_1(z)$, we see that the mode $(-0.2)^k$ does not appear in the output.

9.7 SYSTEMS WITH INPUTS

In the pole-assignment design procedures presented in this chapter, the design resulted in a system in which the initial conditions were driven to zero in some prescribed manner; however, the system had no input. This type of system is called a *regulator control system*. However, in many control systems, it is necessary that the system output $y(t)$ follow, or track, a system input $r(t)$. An example of this type of system is the vertical (or altitude) control system of an aircraft automatic landing system. The aircraft follows a prescribed glide slope, which is a ramp function. Hence the altitude control system input is a ramp function. In this section some techniques will be presented for modifying the pole assignment design method for systems that must track an input.

We first consider the case of low-order single-input single-output systems in which all states are measured; thus no observer is required. The system equations are given by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (9-86)$$

where, for a regulator control system,

$$u(k) = -\mathbf{K}\mathbf{x}(k) \quad (9-87)$$

The control input $u(k)$ is the only variable that we can manipulate; thus this input must then be a function of the system input $r(k)$. Thus we modify $u(k)$ to be a linear function of $r(k)$,

$$u(k) = -\mathbf{K}\mathbf{x}(k) + Nr(k) \quad (9-88)$$

where N is a constant to be determined. Then, from (9-86) and (9-88),

$$\begin{aligned} \mathbf{x}(k+1) &= (\mathbf{A} - \mathbf{BK})\mathbf{x}(k) + \mathbf{B}Nr(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (9-89)$$

Taking the z -transform of these equations and solving for the transfer function yields

$$\frac{Y(z)}{R(z)} = \mathbf{C}(z\mathbf{I} - \mathbf{A} + \mathbf{BK})^{-1} \mathbf{B}N \quad (9-90)$$

Note that the choice of N does not affect the system's relative frequency response, but only the amplitude of the frequency response; the transfer function is independent of N except as a multiplying factor.

A special case is worth noting. For the case that the output is a state, which we can call $x_1(t)$ without loss of generality, we can write the output equation as

$$y(k) = \mathbf{C}\mathbf{x}(k) = [1 \ 0 \ \cdots \ 0]\mathbf{x}(k) = x_1(k)$$

The equation for $u(k)$ can now be expressed as a function of the system error, which is $e(k) = r(k) - y(k)$. From (9-88), we let $N = K_1$, and

$$\begin{aligned} u(k) &= -\mathbf{K}\mathbf{x}(k) + K_1 r(k) \\ &= K_1[r(k) - y(k)] - K_2 x_2(k) - \cdots - K_n x_n(k) \end{aligned} \quad (9-91)$$

Thus the control is a function of the system error, as was the case for classical control design. The system block diagram for this case is given in Figure 9-12. This system is investigated further in Example 9.12.

It is informative to determine the zeros of the system transfer function, (9-90). To locate the zeros, consider the z -transform of (9-89).

$$\begin{aligned} (z\mathbf{I} - \mathbf{A} + \mathbf{BK})\mathbf{X}(z) - \mathbf{B}NR(z) &= 0 \\ \mathbf{C}\mathbf{X}(z) &= Y(z) \end{aligned} \quad (9-92)$$

If $z = z_0$ is a zero of the transfer function, (9-90), then $Y(z_0)$ is zero with $R(z_0)$ and $X(z_0)$ nonzero. Hence we can express (9-92) as

$$\begin{bmatrix} z_0 \mathbf{I} - \mathbf{A} + \mathbf{BK} & -\mathbf{BN} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}(z_0) \\ R(z_0) \end{bmatrix} = 0 \quad (9-93)$$

This equation has a nontrivial solution, since $R(z_0)$ and $X(z_0)$ are not zero. Hence the determinant of the coefficient matrix, which is a polynomial in z_0 , must be zero [7]. The roots of this polynomial are then the zeros of the system transfer function.

The locations of the zeros of the transfer function are not evident from (9-93). However, we can rewrite (9-93) as

$$\begin{bmatrix} z_0 \mathbf{I} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}(z_0) \\ NR(z_0) - \mathbf{KX}(z_0) \end{bmatrix} = 0 \quad (9-94)$$

Here the coefficient matrix is independent of \mathbf{K} and N , and is, in fact, that of the plant alone. Hence the zeros of the closed-loop transfer function are the same as those of the transfer function of the plant and cannot be changed by the design procedure. This design procedure is then of limited value. An example illustrating this design procedure will now be given.

Example 9.12

The system of Example 9.1 will again be considered, but with an input added as in (9-88). For this system the plant state equations are given by

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

The design in Example 9.1 resulted in the gain matrix

$$\mathbf{K} = [4.52 \quad 1.12]$$

Hence

$$\mathbf{BK} = \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} [4.52 \quad 1.12] = \begin{bmatrix} 0.0219 & 0.00542 \\ 0.430 & 0.107 \end{bmatrix}$$

Then

$$z\mathbf{I} - \mathbf{A} + \mathbf{BK} = \begin{bmatrix} z - 0.978 & -0.0898 \\ 0.430 & z - 0.798 \end{bmatrix}$$

and

$$(z\mathbf{I} - \mathbf{A} + \mathbf{BK})^{-1} = \frac{1}{z^2 - 1.776z + 0.819} \begin{bmatrix} z - 0.798 & 0.0898 \\ -0.430 & z - 0.978 \end{bmatrix}$$

The system transfer function is given by (9-90):

$$\frac{Y(z)}{R(z)} = \mathbf{C}(z\mathbf{I} - \mathbf{A} + \mathbf{BK})^{-1} \mathbf{BN} = \frac{(0.00484z + 0.00468)N}{z^2 - 1.776z + 0.819}$$

Note that the system dc gain is given by

$$\left. \frac{Y(z)}{R(z)} \right|_{z=1} = 0.221N$$

The dc gain can be increased by the choice of N . Since $y(k) = x_1(k)$, we can choose $N = K_1$ as in (9-91), and the control will be a function of the error between the input and the output. Since $K_1 = 4.52$, the dc gain is now unity; the system shown in the inner loop of Figure 9-12 is type 1. The step response for this case is shown in Figure 9-13, and the steady-state error is zero.

We will now consider the case that a prediction observer is used to implement the control system. The plant is described by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (9-95)$$

and the observer, from (9-38), is described by

$$[\text{eq. (9-38)}] \quad \mathbf{q}(k+1) = (\mathbf{A} - \mathbf{G}\mathbf{C})\mathbf{q}(k) + \mathbf{G}y(k) + \mathbf{B}u(k)$$

The control is implemented by

$$u(k) = -\mathbf{K}\mathbf{q}(k) \quad (9-96)$$

To be general, we can allow the input to enter both (9-38) and (9-96). These

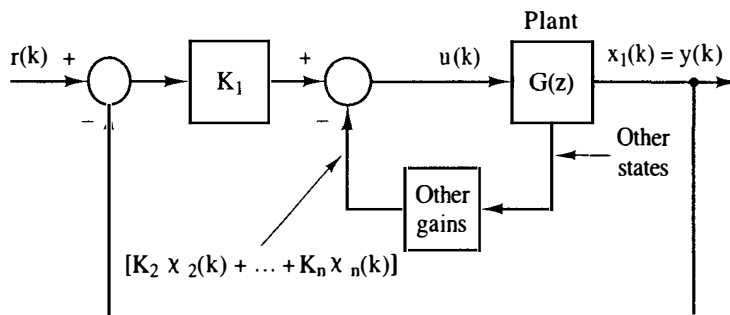


Figure 9-12 System with input.

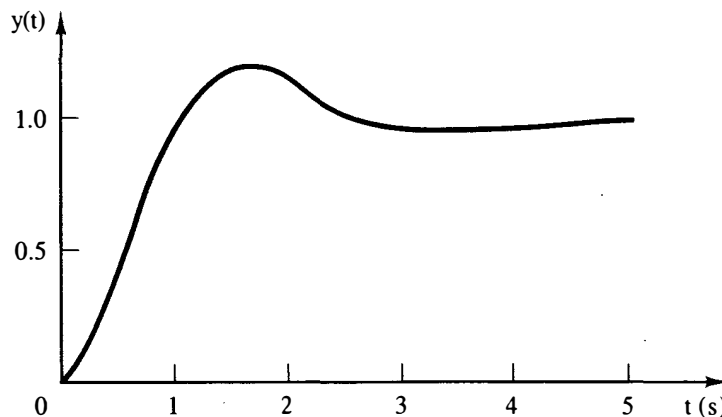


Figure 9-13 Step response for Example 9.12.

equations become

$$\mathbf{q}(k+1) = (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}y(k) + \mathbf{B}u(k) + \mathbf{M}r(k) \quad (9-97)$$

$$u(k) = -\mathbf{K}\mathbf{q}(k) + Nr(k) \quad (9-98)$$

The design problem then is to choose \mathbf{M} and N so as to achieve certain design objectives, such as zeros of the system transfer function, steady-state errors, and so on.

We will consider the case that only the error

$$e(t) = r(t) - y(t) \quad (9-99)$$

is measured, and thus only this signal is available for control. For example, in radar tracking systems, the radar return indicates only the error $e(t)$ between the antenna pointing direction and the direction to the target. The error-control condition can be satisfied in (9-97) and (9-98) by choosing N to be zero and \mathbf{M} equal to $-\mathbf{G}$. Then (9-97) and (9-98) become, respectively,

$$\mathbf{q}(k+1) = (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}[y(k) - r(k)] + \mathbf{B}u(k) \quad (9-100)$$

$$u(k) = -\mathbf{K}\mathbf{q}(k) \quad (9-101)$$

Note that this design procedure also gives no choice in the selection of the system transfer-function zeros. An example will now be given to illustrate this design technique.

Example 9.13

The system of Example 9.12 will be considered, with the observer designed in Example 9.3 to be employed. The plant state equations are given in Example 9.12. The gain matrices for the controller and for the observer are given in Example 9.3. The required matrices are then

$$\mathbf{A} = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix}, \quad \mathbf{C} = [1 \quad 0]$$

$$\mathbf{K} = [4.52 \quad 1.12], \quad \mathbf{G} = \begin{bmatrix} 0.267 \\ 0.0802 \end{bmatrix}$$

Substitution of these matrices into (9-100) and (9-101), and then substitution of (9-101) into (9-100) yields the observer equations

$$\mathbf{q}(k+1) = \begin{bmatrix} 0.711 & 0.0898 \\ -0.510 & 0.798 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.267 \\ 0.0802 \end{bmatrix} [y(k) - r(k)]$$

Note that no design is required beyond that performed in Example 9.3. The system block diagram is given in Figure 9-14. In this block diagram, the controller transfer function is as calculated in Example 9.4, and it is assumed that the input $r(kT)$ is calculated in the computer (as is usually the case). Step responses for this design are given in Figure 9-15. For the response shown by the solid line, the initial conditions of both the plant and the observer are all zero. For the response shown by the dashed line,

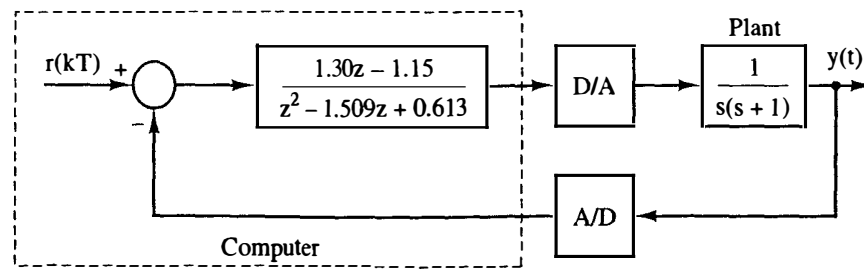


Figure 9-14 System for Example 9.13.

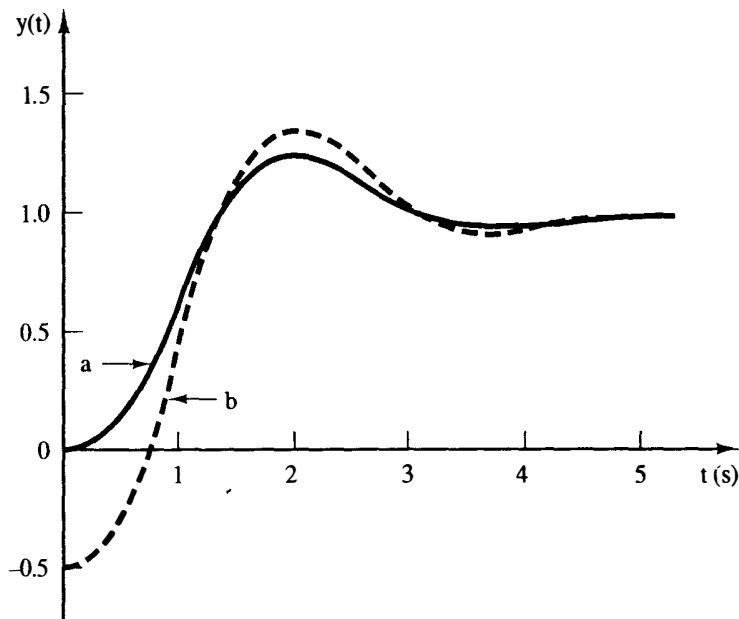


Figure 9-15 Responses for the system of Example 9.13.

the initial conditions are given by

$$\mathbf{x}(0) = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}, \quad \mathbf{q}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

For this design procedure, the transient response appears satisfactory; in addition, the steady-state error is zero.

9.8 SUMMARY

In this chapter a design technique has been presented which is based on the state model of the system, compared to the design methods of Chapter 8, which are based on transfer functions. The design technique presented is the pole-assignment, or pole-placement, technique. This technique allows us to place all the zeros of the system characteristic equation at desirable locations; however, the technique requires that all the system states be known, a condition that at least is generally impractical. To circumvent this condition, an observer is constructed. An observer

estimates the system states, based on the known system inputs and the outputs that can be measured. Then, for the pole-placement design, the estimated states, rather than the actual states, are used to calculate feedback signals.

The design above results in a regulator control system, that is, a control system with no reference input. A method was presented which modifies the pole-placement design for systems that have reference inputs. For the case that the system error is to be used in the control of the system, the modifications required are simple and straightforward.

A problem with observer-based design is that the relative-stability margins at the plant input may not be adequate. This problem can be acute, since the main weakness of analytical design is the inaccuracy of the linear plant model used in the design. In general, a physical system is not linear, and the system's characteristics vary with time, temperature, and so on. Hence an adequate margin of safety is needed, and the relative stability is a measure of that safety margin. In the designs presented in this chapter, the frequency response of the open-loop function of the closed-loop system must be calculated, to ensure that the system has good stability margins.

REFERENCES AND FURTHER READING

1. J. E. Ackermann, "Der Entwurf linearer regelungs Systems in Zustandsraum," *Regelungstech. Prozess-Datenverarb.*, Vol. 7, pp. 297-300, 1972.
2. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1988.
3. D. G. Luenberger, "An Introduction to Observers," *IEEE Trans. Autom. Control*, Vol. AC-16, pp. 596-602, Dec. 1971.
4. Part II, *IEEE Trans. Autom. Control*, Vol. AC-26, Aug. 1981.
5. P. S. Maybeck, *Stochastic Models, Estimation and Control*, Vol. 3. New York: Academic Press, Inc., 1982.
6. J. O'Reilly, *Observers in Linear Systems*. London: Academic Press, Inc. (London) Ltd., 1983.
7. P. M. DeRusso, R. J. Roy, and C. M. Close, *State Variables for Engineers*. New York: John Wiley & Sons, Inc., 1965.
8. J. C. Williams and S. K. Mitter, "Controllability, Observability, Pole Allocation, and State Reconstruction," *IEEE Trans. Autom. Control*, Vol. AC-16, pp. 582-602, Dec. 1971.
9. W. M. Wonham, "On Pole Assignment in Multi-input Controllable Linear Systems," *IEEE Trans. Autom. Control*, Vol. AC-12, pp. 660-665, Dec. 1967.
10. C. T. Leondes and L. M. Novak, "Reduced-Order Observers for Linear Discrete-Time Systems," *IEEE Trans. Autom. Control*, Vol. AC-19, pp. 42-46, Feb. 1974.
11. R. F. Wilson, "An Observer Based Aircraft Automatic Landing System," M.S. thesis, Auburn University, Auburn, AL, 1981.

PROBLEMS

9-1. The plant of Example 9.1 has the state equations

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k)$$

Find the gain matrix \mathbf{K} required to realize the closed-loop characteristic equation with zeros which have a damping ratio, ζ , of 0.46 and a time constant τ of 0.5 s. Use pole-assignment design.

9-2. Consider the plant of Problem 9-1, which has the transfer function, from Example 9.5,

$$G(z) = \frac{z-1}{z} \mathcal{Z} \left[\frac{1}{s^2(s+1)} \right] = \frac{0.00484z + 0.00468}{z^2 - 1.905z + 0.905}$$

- (a) Using the pole assignment design, find the gain matrix \mathbf{K} required to realize the closed-loop characteristic equation with zeros that have the damping ratio $\zeta = 0.707$ and the time constant $\tau = 0.8$ s.
 - (b) To verify the design, show that (9-15) yields the desired closed-loop characteristic equation.
 - (c) Draw a flow graph of the form of Figure 4-20 for the plant. Then add the feedback gains of part (a).
 - (d) Write the state equations for the flow graph of part (c), to verify the system matrix $(\mathbf{A} - \mathbf{BK})$ found in part (b).
 - (e) Find all loop gains in the flow graph of part (c). Then use Mason's formula to verify that the flow graph of part (c) has the characteristic equation of parts (a) and (b).
 - (f) Verify all calculations by computer.
- 9-3. Consider the pole-placement design of Problem 9-2.
- (a) Design a predictor observer for this system, with the time constant equal to one-half the value of Problem 9-2(a) and with the observer critically damped.
 - (b) To check the results of part (a), use (9-46) to show that these results yield the desired observer characteristic equation.
 - (c) Find the control-observer transfer function $D_{ce}(z)$ in Figure 9-8. Use control gain matrix of Problem 9-2(a).
 - (d) The characteristic equation of the closed-loop system of Figure 9-8 is given by

$$1 + D_{ce}(z)G(z) = 0$$

Use $G(z)$ as given and $D_{ce}(z)$ in part (c) to show that this equation yields the same characteristic equation as $\alpha_c(z)\alpha_e(z) = 0$.

- (e) Verify all calculations by computer.
- 9-4. Consider the control system of Problem 9-2.
- (a) Design a reduced-order observer for this system with the time constant equal to one-half the value of Problem 9-2(a).
 - (b) To check the results of part (a), use (9-63) to show that these results yield the desired observer characteristic equation.
 - (c) Find the control-observer transfer function $D_{ce}(z)$ in Figure 9-8. Use the control gain matrix of Problem 9-2(a).
 - (d) The characteristic equation of the closed-loop system of Figure 9-8 is given by

$$1 + D_{ce}(z)G(z) = 0$$

Use $G(z)$ as given and $D_{ce}(z)$ in part (c) to show that this equation yields the same characteristic equation as $\alpha_c(z)\alpha_e(z) = 0$.

(e) Verify all calculations by computer.

9-5. Repeat all parts of Problem 9-3 using a current observer.

9-6. A chamber temperature control system is modeled as shown in Figure P9-6. This system is described in Problem 1-10. For this problem, ignore the disturbance input, $T = 0.6$ s, and let $D(z) = 1$. It was shown in Problem 6-4 that

$$(0.04) \frac{z-1}{z} \left[\frac{2}{s(s+0.5)} \right] = \frac{0.04147}{z-0.7408}$$

Note that the sensor gain is included in this transfer function.

(a) Draw a flow graph of the plant and sensor. Write the state equations with the state variable $x(k)$ equal to the system output and the output $y(k)$ equal to the sensor output.

(b) Find the time constant τ for this closed-loop system.

(c) Using pole-placement design, find the gain K that yield the closed-loop time constant $\tau = 1$ s. Note that the sensor gain does not enter these calculations.

(d) Show that the gain K in part (b) yields the desired closed-loop characteristic equation, using (9-15).

(e) Draw a block diagram for the system that includes the sensor. Let the digital computer realize a gain, K_1 , such that the closed-loop time constant is as given in part (b). The sensor in this system must have the gain given.

(f) Using the characteristic equation for the block diagram of part (e),

$$1 + K_1 G(z)H = 0$$

verify that this block diagram yields the desired characteristic equation.

(g) Verify all calculations by computer.

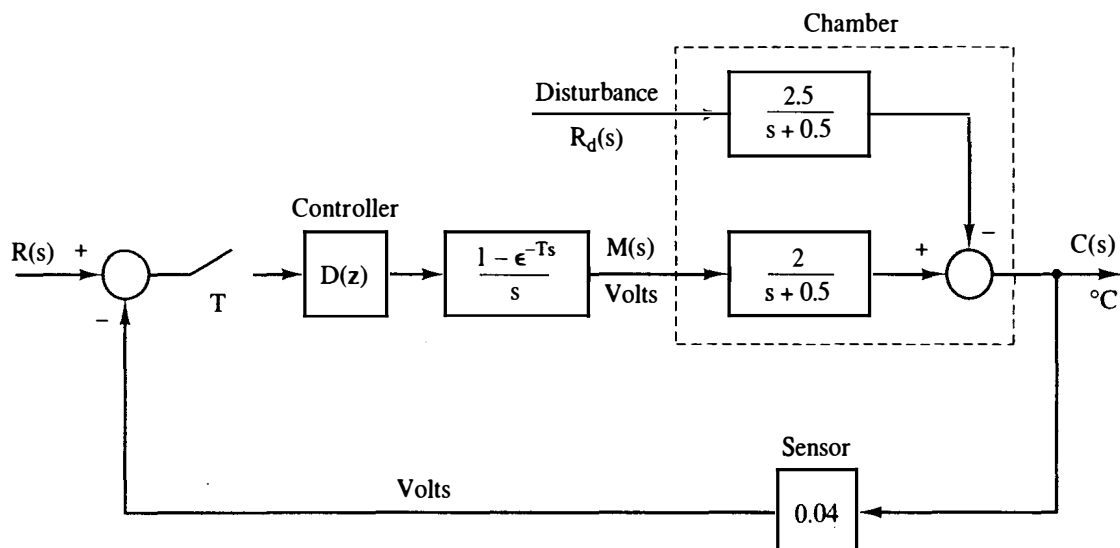


Figure P9-6 Chamber temperature control system.

9-7. Consider the chamber temperature control system of Problem 9-6.

(a) Design a predictor observer for this system, with the time constant equal to one-half the value of Problem 9-6(b).

- (b) To check the results of part (a), use (9-46) to show that these results yield the desired observer characteristic equation.
- (c) Find the control-observer transfer function $D_{ce}(z)$ in Figure 9-8. Use the value of control gain of Problem 9-6(b). Do not include the sensor gain in $D_{ce}(z)$.
- (d) Draw a block diagram of the system in part (b), with $D(z)$ in Figure P9-6 equal to $D_{ce}(z)$.
- (e) Verify the system characteristic equation using the block diagram in part (d).
- (f) What would be the effect of designing a reduced-order estimator for this system?
- (g) Verify all calculations by computer.
- 9-8.** Repeat all parts of Problem 9-7, using a current observer.
- 9-9.** (a) Find the closed-loop state equations for the system of Problem 9-7(a), of the form of (9-56).
 (b) Find the system characteristic equation using the results in part (a), and show that this is the desired equation.
 (c) Repeat parts (a) and (b) for the closed-loop system of Problem 9-8.
 (d) Verify all calculations by computer.
- 9-10.** Consider the chamber temperature control system of Figure P9-6. For this problem, replace the sensor gain $H = 0.04$ with the gain $H = 1$. The system is now a unity-feedback-gain system.
- (a) Work Problem 9-6 with $H = 1$.
 (b) Work Problem 9-7 with $H = 1$.
 (c) Work Problem 9-8 with $H = 1$.
 (d) Work Problem 9-9 with $H = 1$.
- 9-11.** A satellite control system is modeled as shown in Figure P9-11. This system is described in Problem 1.12. For this problem, let $D(z) = 1$. In addition, $K = 1$, $T = 1$ s, $J = 4$, and $H_k = 1$. From the z -transform tables,

$$\frac{z-1}{z} \mathcal{Z} \left[\frac{1}{4s^3} \right] = \frac{0.125(z+1)}{(z-1)^2}$$

A state model for this system is given by

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.125 \\ 0.25 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

where $x_1(k)$ is angular position and $x_2(k)$ is angular velocity.

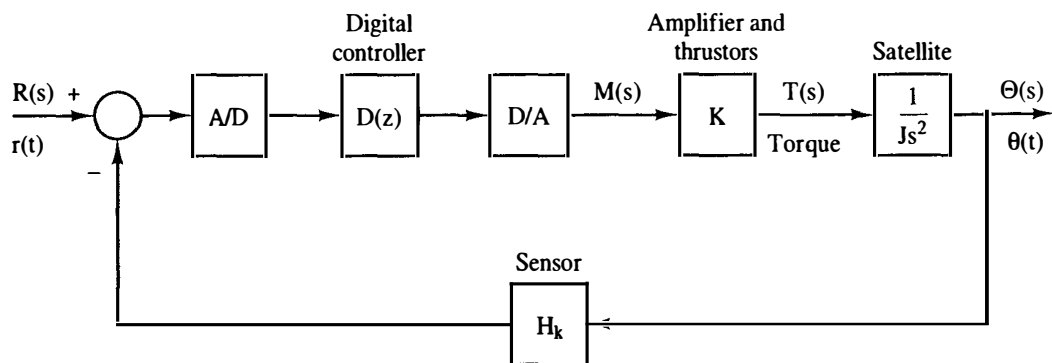


Figure P9-11 Block diagram for a satellite control system.

- (a) Show that the closed-loop system is unstable.
- (b) Using pole-placement design, find the gain matrix \mathbf{K} that yields the closed-loop damping ratio $\zeta = 0.707$ and the time constant $\tau = 4$ s.
- (c) Show that the gain matrix \mathbf{K} in part (b) yields the desired closed-loop characteristic equation, using (9-15).
- (d) Verify all calculations by computer.

9-12. Consider the satellite control system of Problem 9-11.

- (a) Design a predictor observer for this system, with the time constant equal to one-half the value of Problem 9-11(b) and with the observer critically damped.
- (b) To check the results of part (a), use (9-46) to show that these results yield the desired observer characteristic equation.
- (c) Find the control-observer transfer function $D_{ce}(z)$ in Figure 9-8. Use the control gain matrix of Problem 9-11(b), $\mathbf{K} = [0.3893 \quad 1.769]$.
- (d) The characteristic equation of the closed-loop system of Figure 9-8 is given by

$$1 + D_{ce}(z)G(z) = 0$$

Use $G(z)$ as given and $D_{ce}(z)$ in part (c) to show that this equation yields the same characteristic equation as $\alpha_c(z)\alpha_e(z) = 0$.

- (e) Verify all calculations by computer.

9-13. Consider the satellite control system of Problem 9-11.

- (a) Design a reduced-order observer for this system, with the time constant equal to one-half the value of Problem 9-11(b).
- (b) To check the results of part (a), use (9-63) to show that these results yield the desired observer characteristic equation.
- (c) Find the control-observer transfer function $D_{ce}(z)$ in Figure 9-8. Use the control gain matrix of Problem 9-11(b), $\mathbf{K} = [0.3893 \quad 1.769]$.
- (d) The characteristic equation of the closed-loop system of Figure 9-8 is given by

$$1 + D_{ce}(z)G(z) = 0$$

Use $G(z)$ as given and $D_{ce}(z)$ in part (c) to show that this equation yields the same characteristic equation as $\alpha_c(z)\alpha_e(z) = 0$.

- (e) Verify all calculations by computer.

9-14. Repeat all parts of Problem 9-12, using a current observer.

9-15. Consider the reduced-order observer designed in Problem 9-13. In this problem, velocity $[dy/dt]$ is estimated, using position $[y]$ plus other information. We could simply calculate velocity, using one of the numerical differentiators described in Section 8.8. This calculated velocity would then replace the estimate of velocity in the control system. (This is the approach taken in the PID controller.)

- (a) Consider the plant-observer as an open-loop system, as shown in Figure 9-4. Calculate the transfer function $Q(z)/Y(z)$ with $U(z) = 0$, where $q(kT)$ is the observer estimate of velocity. *Hint:* Use (9-62) with $u(k) = 0$.
- (b) Plot the Bode diagram for the transfer function in part (a) versus ω_w .
- (c) One numerical differentiator given in Section 8.8 is

$$D(z) = \frac{z - 1}{Tz}$$

Plot the Bode diagram for this transfer function on the same graph as in part (b).

- (d) Comment on the similarities and differences in the two frequency responses.
- 9-16. Consider that in Figure 9-8, the observer is reduced order, and that the system is single-input single-output. Show that the transfer function $D_{ce}(z)$ of the equivalent controller is given by (9-66).
- 9-17. (a) Show that for the current observer specified in Section 9.5, the transfer matrix from the input $U(z)$ to the estimated states $Q(z)$ is equal to that from the input to the states $X(z)$; that is, show that $Q(z)/U(z) = X(z)/U(z)$.
 (b) Show that for the current observer specified in Section 9.5, the transfer function of the control-observer combination of Figure 9-8 is given by (9-73), which is

$$D_{ce}(z) = zK[zI - A + GCA + BK - GCBK]^{-1}G$$

- 9-18. Consider a system described by (9-82).

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned}$$

For the case that $u(k)$ is not zero, derive the conditions for observability.

- 9-19. Consider the plant of Problem 9-2, which is

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k)$$

Suppose that the output is given by

$$y(k) = [0 \quad 1]\mathbf{x}(k)$$

- (a) Is this system observable?
 (b) Explain the reason for your answer in part (a) in terms of the physical aspects of the system.
- 9-20. Consider the satellite control system of Problem 9-11. Suppose that the output is the measurement of angular velocity, such that

$$y(k) = [0 \quad 1]\mathbf{x}(k)$$

- (a) Is this system observable?
 (b) Explain your answer in part (a) in terms of the physical aspects of the system. We are attempting to estimate position, given velocity.
- 9-21. Consider the temperature control system of Problem 9-6.
 (a) Determine if this system is controllable.
 (b) An observer is added to this system in Problem 9-7, with the equation [see (9-38)]

$$\begin{aligned} q(k+1) &= (A - GC)q(k) + Gy(k) + Bu(k) \\ &= 0.3012q(k) + 10.99y(k) + 1.0368u(k) \end{aligned}$$

Construct a single set of state equations for the plant-observer system, with the state vector $[x(k) \quad q(k)]^T$ and the input $u(k)$ (i.e., there is no feedback).

- (c) Show that this plant-observer system is uncontrollable.
 (d) Based on the results of part (a), which state is uncontrollable?

9-22. Consider the first-order plant described by

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k)$$

and a prediction observer described by (9-38).

(a) Construct a single set of state equations for the plant–observer system, with the state vector $[x(k) \ q(k)]^T$ and the input $u(k)$ (i.e., there is no feedback).

(b) Show that this system is always uncontrollable, regardless of the value of observer gain G .

9-23. Repeat Problem 9-22 for the current observer.

9-24. For the system of Problem 9-22, use a transfer function approach to show that the transfer function $Q(z)/U(z)$ is first order (even though the system is second order) and is equal to $X(z)/U(z)$. Hence the mode of $Q(z)$, which is $(A - GC)^k$, is not excited by the input signal $u(k)$.

9-25. Problem 9-24 is to be repeated for the current observer, with the mode of $Q(z)$ equal to $(A - GCA)^k$.

9-26. In Problem 9-11, a pole-placement design for a satellite control system results in the gain matrix $K = [0.3893 \ 1.769]$. It is desired to have an input signal $r(t)$ applied to the system, so as to realize the system of Figure 9-12. Write the resulting state equations in the form

$$\mathbf{x}(k+1) = \mathbf{A}_1 \mathbf{x}(k) + \mathbf{B}_1 r(k)$$

$$y(k) = \mathbf{C}_1 \mathbf{x}(k)$$

Evaluate all matrices.

9-27. Repeat Problem 9-26 for the temperature control system of Problem 9-6, where the state equations are given by

$$x(k+1) = 0.7408x(k) + 1.0368u(k)$$

$$y(k) = 0.04x(k)$$

and where the control gain is $K = 0.1852$.

9-28. Assume that equation (9-25), Ackermann's formula for pole-assignment design, is the solution of (9-15). Based on this result, show that (9-48), Ackermann's formula for observer design, is the solution of (9-46).

9-29. Given in (9-56) is the closed-loop state model for the pole-placement prediction-estimator design. Extend this model to include plant disturbances and sensor noise, as described in (9-44).

9-30. Derive the closed-loop state model for the pole-placement current-estimator design, given in (9-74).

Linear Quadratic Optimal Control

10.1 INTRODUCTION

In Chapter 8 we presented design techniques that are termed classical, or traditional. The two techniques developed are the frequency-response techniques and the root-locus technique. Both methods are very effective, but are largely trial and error, with experience very useful. Even when an acceptable design is completed, the question remains as to whether a “better” design could be found. The pole-assignment design technique of Chapter 9 is termed a modern technique, and is based on the state-variable model of the plant, rather than on the transfer function as required by the classical methods. In this procedure we assumed that we know the exact locations required for the closed-loop transfer-function poles, and we can realize these locations, at least in the linear model. Of course, for the physical system, the regions in which the pole locations can be placed are limited.

In the pole-assignment technique, we assume that we know the pole locations that yield the “best” control system. In this chapter we develop a different technique that yields the “best” control system. This technique is an optimal design technique, and assumes that we can write a mathematical function which is called the *cost function*. The optimal design procedure minimizes this cost function: hence the term *optimal*. However, in most cases the choice of the cost function involves some trial and error; that is, we are not sure of the exact form that the cost function should take.

For discrete systems, the cost function (also called the *performance index*) is generally the form

$$J_N = \sum_{k=0}^N L[y(k), r(k), u(k)] \quad (10-1)$$

In this relationship k is the sample instant and N is the terminal sample instant. The control system outputs are $y(k)$, the inputs are $r(k)$, and $u(k)$ are the control inputs to the plant.

For a physical system, the control inputs are always constrained. For example, the amplitude of each component of the control vector may be limited, such that

$$|u_i(k)| \leq U_i$$

where each U_i is a given constant, and the subscript i denotes the vector component. For the case of limited control energy, we have

$$u_i^2(k) \leq M_i$$

where each M_i is a given constant. The availability of finite control energy may be represented by a term in the cost function (10-1) as

$$\sum_{k=0}^N \mathbf{u}^T(k) \mathbf{R}(k) \mathbf{u}(k)$$

where $\mathbf{R}(k)$ is a weighting matrix and is not related to $r(k)$. This function is called a *quadratic form*, and will be considered in detail in the following sections. In any case, the control must satisfy certain constraints; any control that satisfies these constraints is called an *admissible control*.

In this chapter the design technique for optimal linear regulator control systems with quadratic cost functions is developed. This design results in a control law of the form

$$\mathbf{u}(k) = -\mathbf{K}(k)\mathbf{x}(k) \quad (10-2)$$

Hence we have linear, time-varying, full-state feedback. The same limitations of full-state feedback apply here as in Chapter 9. It will generally be necessary to utilize an observer to implement the control law. The observer may be designed by the techniques presented in Chapter 9.

The final two topics presented in this chapter are based on the same mathematical foundation as for linear quadratic optimal control. The first of these two topics is a technique for system identification, in which the system transfer function is calculated from input-output data for the physical system. This technique is least-squares system identification, and yields the transfer function that "best" fits the available data. The second topic, Kalman filtering, is an optimal technique for state estimation. Hence the three techniques presented in this chapter are optimal, in the sense that a quadratic, or sum-of-squares, cost function is minimized.

This chapter requires a background in the mathematics of matrices. Appendix IV presents a brief review of the required matrix mathematical definitions and manipulations.

10.2 THE QUADRATIC COST FUNCTION

We begin the development of optimal control design by considering the case of a quadratic cost function of the states and the control signals, that is,

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k) \mathbf{Q}(k) \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R}(k) \mathbf{u}(k) \quad (10-3)$$

where N is finite and $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ are symmetric. The linear plant is described by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k) \mathbf{x}(k) + \mathbf{B}(k) \mathbf{u}(k) \\ y(k) &= \mathbf{C}(k) \mathbf{x}(k) \end{aligned} \quad (10-4)$$

Note that both the plant and the cost-function matrices are allowed to be *time varying*. Recall that the design procedures developed earlier require linear *time-invariant* plants.

The quadratic cost function is considered because the development is simple and the cost function is logical. For example, consider a second-order single-output system with

$$y(k) = \mathbf{C} \mathbf{x}(k) = [c_1 \ c_2] \mathbf{x}(k).$$

Suppose that we want to drive the output to zero, and hence we will choose the cost function to contain $y^2(k)$. Then

$$y^2(k) = y^T(k) y(k) = \mathbf{x}^T(k) \mathbf{C}^T \mathbf{C} \mathbf{x}(k) = \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k)$$

where $\mathbf{Q} = \mathbf{C}^T \mathbf{C}$ is 2×2 . Hence we see for this case that the quadratic function appears naturally.

Consider the quadratic function

$$\begin{aligned} F &= \mathbf{x}^T \mathbf{Q} \mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= q_{11} x_1^2 + (q_{12} + q_{21}) x_1 x_2 + q_{22} x_2^2 \end{aligned} \quad (10-5)$$

Note first that F is a scalar and that there is no loss of generality in assuming \mathbf{Q} to be symmetric. Now, if the quadratic form in (10-5) is positive semidefinite [1], which we will require, then

$$F \geq 0, \quad \mathbf{x} \neq \mathbf{0}$$

$$F = 0, \quad \mathbf{x} = \mathbf{0}$$

Then, in general, minimizing F will minimize the magnitude of states that contribute to F , in some sense. For example, if

$$F = x_1^2 + x_2^2 = \mathbf{x}^T \mathbf{Q} \mathbf{x} = \mathbf{x}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}$$

then minimizing F will tend to minimize the magnitudes of x_1 and x_2 . However, if

$$F = 100x_1^2 + x_2^2 = \mathbf{x}^T \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}$$

then minimizing F will also minimize $|x_1|$ and $|x_2|$, but $|x_1|$ should be much smaller than $|x_2|$. As a third case, suppose that

$$F = x_1^2 = \mathbf{x}^T \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x}$$

(i.e., F is not a function of x_2). Then in minimizing F only x_1 is minimized, and x_2 is determined by its relationship to x_1 .

Consider now the contribution of the control input $\mathbf{u}(k)$. Suppose that

$$\begin{aligned} G &= \mathbf{u}^T \mathbf{R} \mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ &= r_{11}u_1^2 + (r_{12} + r_{21})u_1u_2 + r_{22}u_2^2 \end{aligned} \quad (10-6)$$

Note again that there is no loss in generality in assuming the matrix in a quadratic form is symmetric. If G in (10-6) is positive definite [1], which we will require, then

$$G > 0, \quad \mathbf{u} \neq \mathbf{0}$$

$$G = 0, \quad \mathbf{u} = \mathbf{0}$$

Thus if we minimize G , we are minimizing the control functions. If G were allowed to be only positive semidefinite, some components of the control vector could be quite large when G is minimized.

A mathematical test for positive definiteness is as follows. A square matrix is positive definite if and only if all of its eigenvalues are real and positive. If the eigenvalues are real and positive, except for some that have a value of zero, the matrix is positive semidefinite [1].

Consider the total cost function of (10-3):

$$[\text{eq. (10-3)}] \quad J_N = \sum_{k=0}^N \mathbf{x}^T(k) \mathbf{Q}(k) \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R}(k) \mathbf{u}(k)$$

If $\mathbf{Q}(k)$ were positive semidefinite and $\mathbf{R}(k)$ were the null matrix, minimization of (10-3) would force the $\mathbf{x}(k)$ vector toward zero very quickly, which in general would require a large $\mathbf{u}(k)$. For a physical system, $\mathbf{u}(k)$ is always bounded, and in general the large $\mathbf{u}(k)$ would not be realizable. Hence the positive definite matrix $\mathbf{R}(k)$ is added to the cost function to limit $\mathbf{u}(k)$ to realizable values. Then, in using the cost function (10-3), we include the term involving $\mathbf{x}(k)$, so that, in some sense, the magnitudes of the states are driven toward zero. The term involving $\mathbf{u}(k)$ is included in order that the components of the control vector will be limited in magnitude such that the design is physically realizable.

10.3 THE PRINCIPLE OF OPTIMALITY

The optimal control design problem posed in Section 10.2 may be stated as follows:

For a linear discrete plant described by

$$[\text{eq. (10-4)}] \quad \mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k)$$

determine the control (10-7)

$$\mathbf{u}^o(k) = \mathbf{f}[\mathbf{x}(k)] \quad (10-7)$$

that minimizes the quadratic cost function

$$[\text{eq. (10-3)}] \quad J_N = \sum_{k=0}^N \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k)$$

where N is finite, $\mathbf{Q}(k)$ is positive semidefinite, and $\mathbf{R}(k)$ is positive definite. In (10-7), the superscript o denotes that the control is optimal.

The solution (10-7) can be obtained by several different approaches. The approach to be used here will be through the principle of optimality, developed by Richard Bellman [2,3]. For our purposes, the principle may be stated as follows:

If a closed-loop control $\mathbf{u}^o(k) = \mathbf{f}[\mathbf{x}(k)]$ is optimal over the interval $0 \leq k \leq N$, it is also optimal over *any* subinterval $m \leq k \leq N$, where $0 \leq m \leq N$.

The principle of optimality can be applied as follows. Define the scalar F_k as

$$F_k = \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k) \quad (10-8)$$

Then (10-3) can be expressed as

$$J_N = F_0 + F_1 + \cdots + F_{N-1} + F_N$$

Now, let S_m be the cost from $k = m$ to $k = N$; that is,

$$S_m = J_N - J_{N-m} = F_{N-m+1} + F_{N-m+2} + \cdots + F_{N-1} + F_N \quad (10-9)$$

These cost terms are illustrated in Figure 10-1. Note that k can vary from 0 to N , while m can vary from 1 to $(N+1)$. The principle of optimality states that if J_N is the optimal cost, then so is S_m , for $m = 1, 2, \dots, (N+1)$. We can apply the principle of optimality by first minimizing $S_1 = F_N$, then choosing F_{N-1} to minimize

$$S_2 = F_{N-1} + F_N = S_1^o + F_{N-1}$$

then choosing F_{N-2} to minimize

$$S_3 = F_{N-2} + F_{N-1} + F_N = S_2^o + F_{N-2}$$

and so on, until $S_{N+1} = J_N$ is minimized. Design using this procedure is also known as dynamic programming.

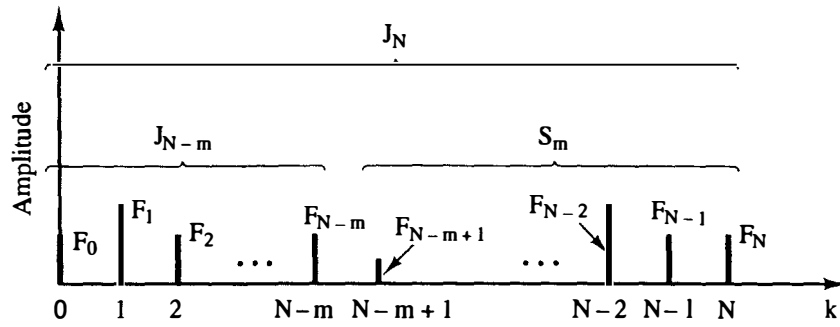


Figure 10-1 Terms in the cost function.

A simple example will now be given to illustrate optimal control design using the principle of optimality. Then the general design procedure will be developed in Section 10.4. Consider the first-order plant described by

$$x(k+1) = 2x(k) + u(k) \quad (10-10)$$

Note that this plant is unstable. We wish to determine the control law $u(k)$ that minimizes

$$J_2 = \sum_{k=0}^2 x^2(k) + u^2(k) = \sum_{k=0}^2 F_k$$

Hence we wish to choose $u(k)$ to minimize

$$J_2 = F_0 + F_1 + F_2 = x^2(0) + u^2(0) + x^2(1) + u^2(1) + x^2(2) + u^2(2) \quad (10-11)$$

subject to the relationship, or *constraint*, (10-10). Note first that $u(2)$ does not affect any of the other terms in (10-11); thus we must choose $u(2) = 0$. This requirement will also appear in the following development.

First, from (10-9),

$$S_1 = J_N - J_{N-1} = J_2 - J_1 = F_2 = x^2(2) + u^2(2)$$

By the principle of optimality, $u(2)$ must minimize this function. Hence

$$\frac{\partial S_1}{\partial u(2)} = 2u(2) = 0$$

since $x(2)$ is independent of $u(2)$. Thus $u(2) = 0$, and

$$S_1^o = x^2(2) \quad (10-12)$$

Next, we calculate S_2 .

$$S_2 = S_1^o + F_1 = S_1^o + x^2(1) + u^2(1)$$

From (10-10) and (10-12), S_2 may be expressed as

$$S_2 = [2x(1) + u(1)]^2 + x^2(1) + u^2(1)$$

By the principle of optimality,

$$\frac{\partial S_2}{\partial u(1)} = 2[2x(1) + u(1)] + 2u(1) = 0$$

since $x(1)$ is independent of $u(1)$. Thus

$$u(1) = -x(1)$$

and

$$S_2^o = 3x^2(1)$$

By the same procedure,

$$S_3 = J_2 = S_2^o + F_2 = S_2^o + x^2(0) + u^2(0)$$

Hence

$$\begin{aligned} S_3 &= 3x^2(1) + x^2(0) + u^2(0) \\ &= 3[2x(0) + u(0)]^2 + x^2(0) + u^2(0) \end{aligned}$$

and

$$\frac{\partial S_3}{\partial u(0)} = 6[2x(0) + u(0)] + 2u(0) = 0$$

Thus

$$u(0) = -1.5x(0)$$

Therefore, the optimal control sequence is given by

$$\begin{aligned} u(0) &= -1.5x(0) \\ u(1) &= -x(1) \\ u(2) &= 0 \end{aligned} \tag{10-13}$$

The minimum-cost function is then

$$\begin{aligned} S_3^o &= J_2^o = 3[0.5x(0)]^2 + x^2(0) + [-1.5x(0)]^2 \\ &= 4x^2(0) \end{aligned}$$

Thus *no* choice of $\{u(0), u(1), u(2)\}$ will yield a smaller value of J_2 than will that of (10-13).

The following points should be made about the example above. First, while no assumption was made concerning the form of the control law, the optimal control law is linear and of the form

$$u(k) = -K(k)x(k)$$

Next, even though the plant is time invariant, the feedback gains required, $K(k)$, are time varying. In the following section a general solution to this optimal control

design problem will be derived, and the points above will be shown to also apply to the general case.

Note also that the optimal control design is solved in reverse time. The optimal gain $K(i)$ cannot be calculated until all the remaining $K(j), i < j \leq N$ are known.

10.4 LINEAR QUADRATIC OPTIMAL CONTROL

We will now solve the linear quadratic (LQ) optimal design problem posed in Section 10.3, which is as follows:

For a linear discrete plant described by

$$\begin{aligned} \text{[eq. (10-4)]} \quad \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) \end{aligned}$$

determine the control

$$\text{[eq. (10-7)]} \quad \mathbf{u}^o(k) = \mathbf{f}[\mathbf{x}(k)]$$

that minimizes the quadratic cost function

$$\text{[eq. (10-3)]} \quad J_N = \sum_{k=0}^N \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k)$$

where N is finite, $\mathbf{Q}(k)$ is positive semidefinite, and $\mathbf{R}(k)$ is positive definite.

First a short review of the differentiation of quadratic functions is in order (see Appendix IV). Given the quadratic function $\mathbf{x}^T \mathbf{Q} \mathbf{x}$, then

$$\frac{\partial}{\partial \mathbf{x}} [\mathbf{x}^T \mathbf{Q} \mathbf{x}] = 2\mathbf{Q}\mathbf{x} \quad (10-14)$$

Given the bilinear form $\mathbf{x}^T \mathbf{Q} \mathbf{y}$, then

$$\frac{\partial}{\partial \mathbf{x}} [\mathbf{x}^T \mathbf{Q} \mathbf{y}] = \mathbf{Q} \mathbf{y} \quad (10-15)$$

and

$$\frac{\partial}{\partial \mathbf{y}} [\mathbf{x}^T \mathbf{Q} \mathbf{y}] = \mathbf{Q}^T \mathbf{x} \quad (10-16)$$

In (10-14) we can assume that \mathbf{Q} is symmetric. However, in (10-15) and (10-16), \mathbf{Q} is not necessarily symmetric. Note also that since $\mathbf{x}^T \mathbf{Q} \mathbf{y}$ is a scalar,

$$\mathbf{x}^T \mathbf{Q} \mathbf{y} = (\mathbf{x}^T \mathbf{Q} \mathbf{y})^T = \mathbf{y}^T \mathbf{Q}^T \mathbf{x}$$

In (10-9) we defined S_m to be the cost from the $(N - m)$ sample instant to the N (terminal) sample instant, that is,

$$\text{[eq. (10-9)]} \quad S_m = J_N - J_{N-m}$$

Hence from Figure 10-1 we see that S_{m+1} can be expressed as

$$S_{m+1} = S_m + F_{N-m}$$

From (10-8), we can then write

$$\begin{aligned} S_{m+1} = S_m &+ \mathbf{x}^T(N-m)\mathbf{Q}(N-m)\mathbf{x}(N-m) \\ &+ \mathbf{u}^T(N-m)\mathbf{R}(N-m)\mathbf{u}(N-m) \end{aligned} \quad (10-17)$$

Thus, from the principle of optimality,

$$\frac{\partial S_{m+1}}{\partial \mathbf{u}(N-m)} = \mathbf{0} \quad (10-18)$$

if S_m in (10-17) is replaced with S_m^o .

Consider first S_1 :

$$S_1 = F_N = \mathbf{x}^T(N)\mathbf{Q}(N)\mathbf{x}(N) + \mathbf{u}^T(N)\mathbf{R}(N)\mathbf{u}(N) \quad (10-19)$$

Since $\mathbf{x}(N)$ is independent of $\mathbf{u}(N)$, $\mathbf{u}^o(N) = \mathbf{0}$ and

$$S_1^o = \mathbf{x}(N)^T \mathbf{Q}(N) \mathbf{x}(N) \quad (10-20)$$

We see then that S_1^o is quadratic in $\mathbf{x}(N)$.

For S_2 , we get the expression, from (10-17), that

$$S_2 = S_1^o + \mathbf{x}^T(N-1)\mathbf{Q}(N-1)\mathbf{x}(N-1) + \mathbf{u}^T(N-1)\mathbf{R}(N-1)\mathbf{u}(N-1) \quad (10-21)$$

where, from (10-4) and (10-20),

$$S_1^o = [\mathbf{A}\mathbf{x}(N-1) + \mathbf{B}\mathbf{u}(N-1)]^T \mathbf{Q}(N) [\mathbf{A}\mathbf{x}(N-1) + \mathbf{B}\mathbf{u}(N-1)]|_{\mathbf{u}^o(N-1)} \quad (10-22)$$

With (10-22) substituted into (10-21), we solve

$$\frac{\partial S_2}{\partial \mathbf{u}(N-1)} = \mathbf{0}$$

for $\mathbf{u}(N-1)$, with the result of the form

$$\mathbf{u}^o(N-1) = -\mathbf{K}(N-1)\mathbf{x}(N-1) \quad (10-23)$$

The derivation of this result is straightforward and is given as Problem 10-1. We will not at this time solve for $\mathbf{K}(N-1)$. However, when (10-23) is substituted into (10-21), the result is seen to be quadratic in $\mathbf{x}(N-1)$. Hence we can write

$$S_2^o = \mathbf{x}^T(N-1)\mathbf{P}(N-1)\mathbf{x}(N-1) \quad (10-24)$$

where $\mathbf{P}(N-1)$ is symmetric.

Following the development in the paragraph above in solving for S_3 , we see that S_3^o is also quadratic. Hence we may write

$$S_3^o = \mathbf{x}^T(N-2)\mathbf{P}(N-2)\mathbf{x}(N-2) \quad (10-25)$$

Thus the general relationship of S_m^o is seen to be

$$S_m^o = \mathbf{x}^T(N - m + 1)\mathbf{P}(N - m + 1)\mathbf{x}(N - m + 1) \quad (10-26)$$

From (10-4), we can then express S_m^o as

$$S_m^o = [\mathbf{Ax}(N - m) + \mathbf{Bu}(N - m)]^T \mathbf{P}(N - m + 1)[\mathbf{Ax}(N - m) + \mathbf{Bu}(N - m)] \quad (10-27)$$

Next we substitute (10-27) into (10-17) to get the expression for S_{m+1} . Then we differentiate S_{m+1} with respect to $\mathbf{u}(N - m)$, set this to zero, and solve for $\mathbf{u}^o(N - m)$. The algebra becomes unwieldy, and to simplify this, we will drop the notational dependence of $(N - m)$. Then, in the final solution, we will reinsert this dependence.

From (10-27) and (10-17),

$$S_{m+1} = [\mathbf{Ax} + \mathbf{Bu}]^T \mathbf{P}(N - m + 1)[\mathbf{Ax} + \mathbf{Bu}] + \mathbf{x}^T \mathbf{Qx} + \mathbf{u}^T \mathbf{Ru} \quad (10-28)$$

Then, from (10-14), (10-15), and (10-16),

$$\frac{\partial S_{m+1}}{\partial \mathbf{u}} = \mathbf{B}^T \mathbf{P}(N - m + 1)[\mathbf{Ax} + \mathbf{Bu}] + \mathbf{B}^T \mathbf{P}(N - m + 1)[\mathbf{Ax} + \mathbf{Bu}] + 2\mathbf{Ru} = \mathbf{0}$$

or

$$2\mathbf{B}^T \mathbf{P}(N - m + 1)\mathbf{Ax} + 2[\mathbf{B}^T \mathbf{P}(N - m + 1)\mathbf{B} + \mathbf{R}]\mathbf{u} = \mathbf{0}$$

Thus the desired solution is

$$\mathbf{u}^o = -[\mathbf{B}^T \mathbf{P}(N - m + 1)\mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(N - m + 1)\mathbf{Ax} \quad (10-29)$$

Hence, from (10-23) and (10-29), the optimal gain matrix is

$$\begin{aligned} \mathbf{K}(N - m) &= [\mathbf{B}^T(N - m)\mathbf{P}(N - m + 1)\mathbf{B}(N - m) + \mathbf{R}(N - m)]^{-1} \\ &\quad \times \mathbf{B}^T(N - m)\mathbf{P}(N - m + 1)\mathbf{A}(N - m) \end{aligned} \quad (10-30)$$

and

$$\mathbf{u}^o(N - m) = -\mathbf{K}(N - m)\mathbf{x}(N - m) \quad (10-31)$$

Next we develop the expression for S_{m+1}^o . In (10-28), from (10-31),

$$\mathbf{Ax} + \mathbf{Bu} = [\mathbf{A} - \mathbf{BK}]\mathbf{x}$$

and

$$S_{m+1}^o = \{[\mathbf{A} - \mathbf{BK}]\mathbf{x}\}^T \mathbf{P}(N - m + 1)[\mathbf{A} - \mathbf{BK}]\mathbf{x} + \mathbf{x}^T \mathbf{Qx} + [\mathbf{Kx}]^T \mathbf{RKx}$$

or

$$S_{m+1}^o = \mathbf{x}^T \{[\mathbf{A} - \mathbf{BK}]^T \mathbf{P}(N - m + 1)[\mathbf{A} - \mathbf{BK}] + \mathbf{Q} + \mathbf{K}^T \mathbf{RK}\} \mathbf{x}$$

But, from (10-26),

$$S_{m+1}^o = \mathbf{x}^T \mathbf{P}(N - m)\mathbf{x}$$

Hence

$$\begin{aligned} \mathbf{P}(N-m) &= [\mathbf{A}(N-m) - \mathbf{B}(N-m)\mathbf{K}(N-m)]^T \mathbf{P}(N-m+1) \\ &\quad \times [\mathbf{A}(N-m) - \mathbf{B}(N-m)\mathbf{K}(N-m)] \\ &\quad + \mathbf{Q}(N-m) + \mathbf{K}^T(N-m)\mathbf{R}(N-m)\mathbf{K}(N-m) \end{aligned} \quad (10-32)$$

The final design equations will now be summarized. The design progresses backward in time from $k = N$. We know the final value of the \mathbf{P} matrix from (10-20) and (10-26).

$$\mathbf{P}(N) = \mathbf{Q}(N) \quad (10-33)$$

For a time-invariant system and cost function, from (10-30) we obtain the optimal gain-matrix expression

$$\mathbf{K}(N-m) = [\mathbf{B}^T \mathbf{P}(N-m+1) \mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(N-m+1) \mathbf{A} \quad (10-34)$$

Hence we can solve for $\mathbf{K}(N-1)$, and from (10-32),

$$\begin{aligned} \mathbf{P}(N-m) &= [\mathbf{A} - \mathbf{B}\mathbf{K}(N-m)]^T \mathbf{P}(N-m+1) [\mathbf{A} - \mathbf{B}\mathbf{K}(N-m)] \\ &\quad + \mathbf{Q} + \mathbf{K}^T(N-m)\mathbf{R}\mathbf{K}(N-m) \end{aligned} \quad (10-35)$$

This equation can be expressed in other forms (see Problem 10-2). One of the simpler forms is

$$\mathbf{P}(N-m) = \mathbf{A}^T \mathbf{P}(N-m+1) [\mathbf{A} - \mathbf{B}\mathbf{K}(N-m)] + \mathbf{Q} \quad (10-36)$$

Thus we may solve either (10-35) or (10-36) for $\mathbf{P}(N-1)$. We see then that (10-34) and (10-35) or (10-36) form a set of nonlinear difference equations which may be solved recursively for $\mathbf{K}(N-m)$ and $\mathbf{P}(N-m)$. Thus (10-33), (10-34), and (10-36) form the design equations.

We can also express the design equations as a function of k . In (10-31), (10-34), and (10-36), let $k = N - m$. The design equations are, for $k = 0, 1, 2, \dots, N-1$,

$$\mathbf{u}^o(k) = -\mathbf{K}(k)\mathbf{x}(k) \quad (10-37)$$

$$\mathbf{K}(k) = [\mathbf{B}^T \mathbf{P}(k+1) \mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A} \quad (10-38)$$

$$\mathbf{P}(k) = \mathbf{A}^T \mathbf{P}(k+1) [\mathbf{A} - \mathbf{B}\mathbf{K}(k)] + \mathbf{Q} \quad (10-39)$$

with $\mathbf{P}(N) = \mathbf{Q}$ and $\mathbf{K}(N) = \mathbf{0}$.

Note from (10-9) that J_N can be expressed as

$$J_N = S_N + J_0 = S_N + \mathbf{x}^T(0)\mathbf{Q}(0)\mathbf{x}(0) + \mathbf{u}^T(0)\mathbf{R}(0)\mathbf{u}(0)$$

since $F_0 = J_0$. But from Figure 10-1, this expression is simply S_{N+1} . Thus from (10-26), the minimum cost is given by

$$J_N^o = S_{N+1}^o = \mathbf{x}^T(0)\mathbf{P}(0)\mathbf{x}(0) \quad (10-40)$$

For the case that the system and/or the cost function are time varying, the

following substitutions must be made in (10-38) and (10-39):

$$\begin{aligned} \mathbf{A} &\rightarrow \mathbf{A}(k) \\ \mathbf{B} &\rightarrow \mathbf{B}(k) \\ \mathbf{Q} &\rightarrow \mathbf{Q}(k) \\ \mathbf{R} &\rightarrow \mathbf{R}(k) \end{aligned} \tag{10-41}$$

Two examples illustrating this design procedure will now be given.

Example 10.1

We wish to design an optimal control law for the system

$$x(k+1) = 2x(k) + u(k)$$

with the cost function

$$J_2 = \sum_{k=0}^2 x^2(k) + u^2(k)$$

Note that this is the same design problem considered in Section 10.3. Now, the required parameters are

$$\begin{aligned} A &= 2 & Q &= 1 \\ B &= 1 & R &= 1 \end{aligned}$$

From (10-33),

$$P(2) = Q(2) = Q = 1$$

From (10-38), the gain required is

$$\begin{aligned} K(2-1) &= K(1) = [B^T P(2)B + R]^{-1} B^T P(2)A \\ &= [1 + 1]^{-1}(1)(1)(2) = 1 \end{aligned}$$

From (10-39),

$$\begin{aligned} P(1) &= A^T P(2)\{A - BK(1)\} + Q \\ &= 2(1)\{2 - (1)(1)\} + 1 = 3 \end{aligned}$$

Then, from (10-38),

$$\begin{aligned} K(0) &= [B^T P(1)B + R]^{-1} B^T P(1)A \\ &= [3 + 1]^{-1}(1)(3)(2) = 1.5 \end{aligned}$$

and from (10-39),

$$P(0) = 2(3)\{2 - 1.5\} + 1 = 4$$

Hence the optimal gain schedule is

$$\{K(0), K(1)\} = \{1.5, 1\}$$

and from (10-40), the minimum cost is

$$J_2^o = x^T(0)P(0)x(0) = 4x^2(0)$$

The results check those obtained in Section 10.3.

Example 10.2



As a second example, we consider the servo system utilized in several examples of pole-assignment design in Chapter 9. The system is shown in Figure 10-2, and has the state model (see Example 9.1)

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

where $x_1(t)$ is the shaft angle and $x_2(t)$ is the shaft angular velocity. We choose the cost function to be

$$J_2 = \sum_{k=0}^2 \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) + R u^2(k)$$

with

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = 1$$

Thus velocity is ignored in the cost function. We are attempting to minimize the magnitude of the position $x_1(k)$ without regard to the velocity $x_2(k)$ required. The weight of the control, R , in the cost function is chosen arbitrarily. N is chosen equal to 2 so that the solution can be calculated by hand. Thus, from (10-33),

$$\mathbf{P}(2) = \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

In (10-38), for $k = 1$,

$$\begin{aligned} \mathbf{B}^T \mathbf{P}(2) \mathbf{B} + R &= [0.00484 \quad 0.0952] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} + 1 \\ &= [0.00484 \quad 0] \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} + 1 = 1.0000234 \end{aligned}$$

Then, in (10-38),

$$\begin{aligned} \mathbf{K}(1) &= [\mathbf{B}^T \mathbf{P}(2) \mathbf{B} + R]^{-1} \mathbf{B}^T \mathbf{P}(2) \mathbf{A} \\ &= \frac{1}{1.0000234} [0.00484 \quad 0.0952] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \\ &= [0.00484 \quad 0.000461] \end{aligned}$$

To calculate $\mathbf{P}(1)$ from (10-39), we need

$$\begin{aligned} \mathbf{B} \mathbf{K}(1) &= \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} [0.00484 \quad 0.000461] \\ &= \begin{bmatrix} 2.34 \times 10^{-5} & 2.23 \times 10^{-6} \\ 4.61 \times 10^{-4} & 4.39 \times 10^{-5} \end{bmatrix} \end{aligned}$$

Then

$$[\mathbf{A} - \mathbf{B} \mathbf{K}(1)] = \begin{bmatrix} 1.00 & 0.0952 \\ -4.61 \times 10^{-4} & 0.905 \end{bmatrix}$$

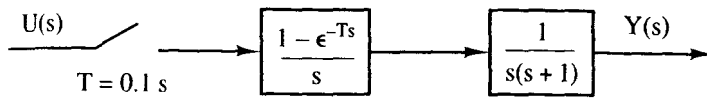


Figure 10-2 Servomotor system.

and

$$\mathbf{A}^T \mathbf{P}(2) [\mathbf{A} - \mathbf{BK}(1)] = \begin{bmatrix} 1 & 0.0952 \\ 0.0952 & 0.00906 \end{bmatrix}$$

Then, from (10-39),

$$\begin{aligned} \mathbf{P}(1) &= \mathbf{A}^T \mathbf{P}(2) [\mathbf{A} - \mathbf{BK}(1)] + \mathbf{Q} \\ &= \begin{bmatrix} 2.0 & 0.0952 \\ 0.0952 & 0.00906 \end{bmatrix} \end{aligned}$$

The calculations above are carried through in detail to illustrate the calculations required for one step of the solution of the difference equations (10-38) and (10-39) for a second-order system with a single input. Obviously, a computer solution is required. If these difference equations are solved again, the results are

$$\mathbf{K}(0) = [0.0187 \quad 0.00298]$$

and

$$\mathbf{P}(0) = \begin{bmatrix} 3.0 & 0.276 \\ 0.276 & 0.0419 \end{bmatrix}$$

A MATLAB program that performs the calculations of this example is given by

```
format short e
A = [1 0.0952; 0 0.905];
B = [0.00484; 0.0952];
Q = [1 0; 0 0];
R = 1;
N = 2;
P = Q;
disp('    k    Gains')
for n=1:N
    KK = inv(B'*P*B + R)*B'*P*A;
    P1=A'*P*(A - B*KK) + Q;
    P=P1;
    k = N-n+1;
    [k, KK]
end
disp(' The final value of the P matrix is:')
P
```

Example 10.3

Next a more practical design will be performed, using a computer solution; that is, (10-38) and (10-39) are solved recursively via the computer. We will compare the results

to those of Example 9.1. In Example 9.1 the plant equations are (see Figure 10-2)

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k)$$

and the closed-loop characteristic equation zeros had a time constant of 1.0 s. We generally consider the response to have settled out in five time constants, or 5 s in this case. Hence we will choose N of the cost function of (10-3) to be 51 and compare responses for the first 5 s. The \mathbf{Q} and \mathbf{R} matrices of (10-3) are chosen to be the same as in Example 10.2, that is,

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = 1$$

The optimal gains obtained for this design are plotted in Figure 10-3. Also shown are the optimal gains for values of R of 0.1 and 0.03, with \mathbf{Q} unchanged. As the value of R is reduced, the contribution of the control effort to the cost function is reduced. Hence the control effort will be increased in order to reduce the magnitudes of the states. This effect is seen in Figure 10-3.

Shown in Figure 10-4 is the initial-condition response, for $x_1(t)$, of this system for $R = 1.0$ and $R = 0.03$, with

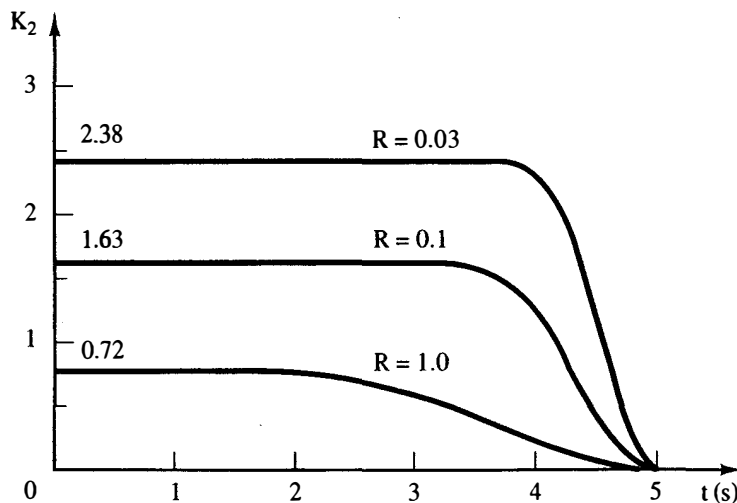
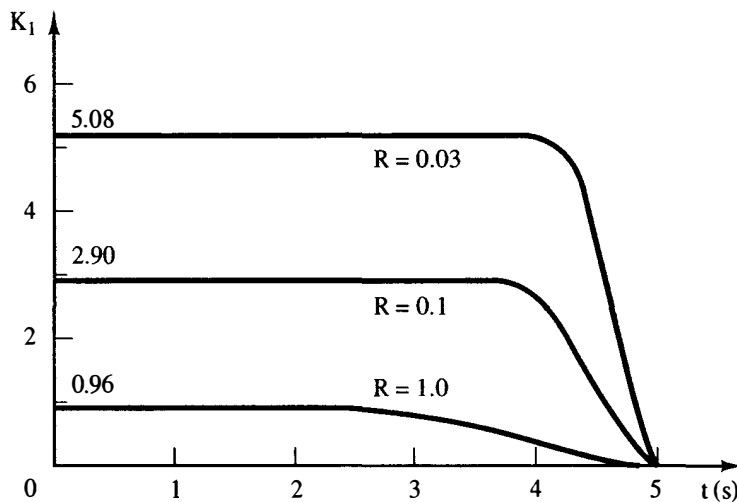


Figure 10-3 Optimal gains for Example 10.3.

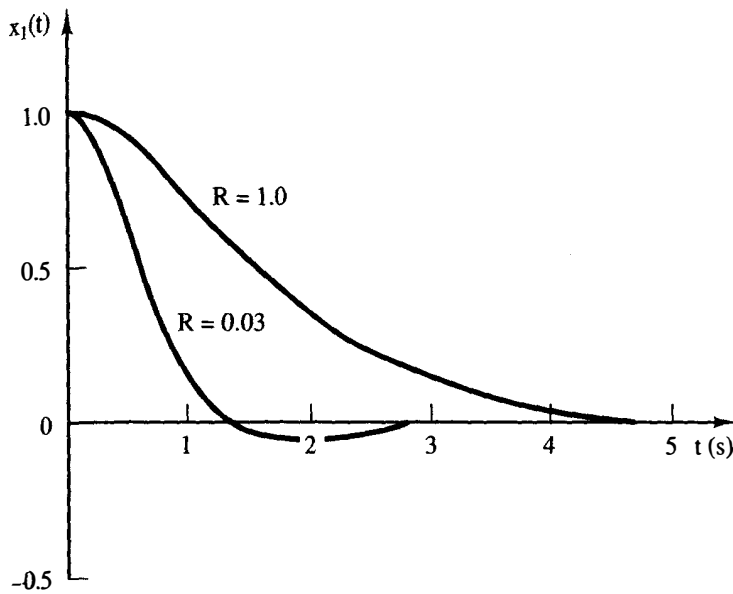


Figure 10-4 Time response for Example 10.3.

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

The effects of the choice of R on the transient response of the closed-loop system can be seen from this figure. Also, by comparing Figures 10-3 and 10-4, we see that the response has reached zero before the gains begin decreasing for the case that $R = 0.03$. This is not true for the case that $R = 1.0$.

The initial-condition response for Example 9.1 is given in Figure 9-2. It is seen that the optimal system response settles faster for $R = 0.03$, but slower for $R = 1.0$.

10.5 THE MINIMUM PRINCIPLE

In the developments above, Bellman's principle was used in deriving the optimal control design equations (10-38) and (10-39). The equations may also be derived utilizing the minimum principle [4]. The minimum principle is presented in this section. This principle will prove to be useful when steady-state optimal control is considered in the following section.

The optimal control problem posed in Section 10.2 may be stated as follows:

For a linear discrete plant described by

$$[\text{eq. (10-4)}] \quad \mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k)$$

determine the control

$$[\text{eq. (10-7)}] \quad \mathbf{u}^o(k) = \mathbf{f}[\mathbf{x}(k)]$$

that minimizes the quadratic cost function

$$[\text{eq. (10-3)}] \quad J_N = \frac{1}{2} \sum_{k=0}^N \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k)$$

where N is finite, $\mathbf{Q}(k)$ is positive semidefinite, and $\mathbf{R}(k)$ is positive definite. In (10-7), the superscript o denotes that the control is optimal. Note that the factor $\frac{1}{2}$ added to (10-3) will not change the results, but will simplify the following derivations somewhat.

The minimum principle may be utilized to solve the optimal control problem.

Minimum Principle [4]. If the input $\mathbf{u}^o(k)$ and the corresponding trajectory $\mathbf{x}^o(k)$ are optimal, there exists a nontrivial vector sequence $\{\mathbf{p}^o(k)\}$ such that $\mathbf{u}^o(k)$ is the value of $\mathbf{u}(k)$ that minimizes the Hamiltonian

$$H = \frac{1}{2}[\mathbf{x}^{oT}(k)\mathbf{Q}(k)\mathbf{x}^o(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k)] + [\mathbf{p}^o(k+1)]^T[\mathbf{A}(k)\mathbf{x}^o(k) + \mathbf{B}(k)\mathbf{u}(k)] \quad (10-42)$$

and the "costate vector" $\mathbf{p}^o(k)$ satisfies

$$\mathbf{p}^o(k) = \frac{\partial H}{\partial \mathbf{x}^o(k)}, \quad \mathbf{p}^o(N) = \mathbf{Q}(N)\mathbf{x}^o(N) \quad (10-43)$$

for $k \leq N$.

From (10-43) we obtain

$$\mathbf{p}^o(k) = \mathbf{Q}(k)\mathbf{x}^o(k) + \mathbf{A}^T(k)\mathbf{p}^o(k+1) \quad (10-44)$$

and $\partial H / \partial \mathbf{u}(k) = 0$ yields

$$\mathbf{u}^o(k) = -\mathbf{R}^{-1}(k)\mathbf{B}^T(k)\mathbf{p}^o(k+1) \quad (10-45)$$

Hence this approach leads to the state equations (10-4),

$$[\text{eq. (10-4)}] \quad \mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k)$$

and the costate equations (10-44), with the optimal control given by (10-45). The state equations will have an initial condition $\mathbf{x}(k_o)$, and are solved forward in time. The costate equations have the final state (10-43), and are solved backward in time. However, since the state and costate equations are coupled, these equations constitute a linear two-point boundary value problem. It is beyond the scope of this book to discuss the solution of two-point boundary value problems. However, it can be shown that the solution of (10-4), (10-44), and (10-45) yields the same optimal gain matrix as does the solution of (10-38) and (10-39).

10.6 STEADY-STATE OPTIMAL CONTROL

If we consider the design results of Example 10.3, given in Figure 10-3, we see that the optimal gains approach constant values with decreasing time. In general, this is true for time-invariant systems and cost functions.

Note also that if we increase N in Example 10.3, the optimal gains for the final 51 sample instants are the same as those calculated in this example. The values of

the gains at earlier sampling instants will simply be those constant values indicated in Figure 10-3. Hence if we allow N to approach infinity, or if we allow the initial time to approach $-\infty$, we obtain a steady-state solution in which the optimal gains are constant values. We consider this steady-state solution in this section.

First we consider the difference equations (10-38) and (10-39) employed in the design of optimal control systems.

$$[\text{eq. (10-38)}] \quad \mathbf{K}(k) = [\mathbf{B}^T \mathbf{P}(k+1) \mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A}$$

$$[\text{eq. (10-39)}] \quad \mathbf{P}(k) = \mathbf{A}^T \mathbf{P}(k+1) [\mathbf{A} - \mathbf{B} \mathbf{K}(k)] + \mathbf{Q}$$

In this section \mathbf{A} , \mathbf{B} , \mathbf{Q} , and \mathbf{R} are assumed to be *constant*. If (10-38) is substituted into (10-39), the result is a difference equation for the matrix \mathbf{P} .

$$\begin{aligned} \mathbf{P}(k) = & \mathbf{A}^T \mathbf{P}(k+1) \{ \mathbf{A} - \mathbf{B} [\mathbf{B}^T \mathbf{P}(k+1) \mathbf{B} + \mathbf{R}]^{-1} \\ & \times \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A} \} + \mathbf{Q} \end{aligned} \quad (10-46)$$

This equation is often written in a slightly different form,

$$\begin{aligned} \mathbf{P}(k) = & \mathbf{A}^T \mathbf{P}(k+1) \mathbf{A} + \mathbf{Q} - \mathbf{A}^T \mathbf{P}(k+1) \\ & \times \mathbf{B} [\mathbf{B}^T \mathbf{P}(k+1) \mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A} \end{aligned} \quad (10-47)$$

and is referred to as the *discrete Riccati equation*. The inverse in (10-47) always exists since \mathbf{R} is positive definite and $\mathbf{P}(k+1)$ is at least positive semidefinite [the cost function in (10-3) and (10-26) cannot be negative].

Before deriving the steady-state solution to (10-47), we will derive a non-recursive solution for $\mathbf{P}(k)$ in the discrete Riccati equation for time-invariant systems. This solution will then yield the steady-state solution. We can express the state and costate equations of (10-4) and (10-44) as

$$\begin{bmatrix} \mathbf{x}^o(k) \\ \mathbf{p}^o(k) \end{bmatrix} = \mathcal{H} \begin{bmatrix} \mathbf{x}^o(k+1) \\ \mathbf{p}^o(k+1) \end{bmatrix} \quad (10-48)$$

where

$$\mathcal{H} = \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{A}^{-1} \mathbf{R}_C \\ \mathbf{Q} \mathbf{A}^{-1} & \mathbf{A}^T + \mathbf{Q} \mathbf{A}^{-1} \mathbf{R}_C \end{bmatrix} \quad (10-49)$$

and

$$\mathbf{R}_C = \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \quad (10-50)$$

Note that \mathcal{H} is $2n \times 2n$. For a matrix

$$\mathcal{H} = \begin{bmatrix} \mathbf{D} & \mathbf{E} \\ \mathbf{F} & \mathbf{G} \end{bmatrix} \quad (10-51)$$

where the partitions \mathbf{D} , \mathbf{E} , \mathbf{F} , and \mathbf{G} are $n \times n$, then

$$\mathcal{H}^{-1} = \begin{bmatrix} \mathbf{D}^{-1} + \mathbf{D}^{-1} \mathbf{E} [\mathbf{G} - \mathbf{F} \mathbf{D}^{-1} \mathbf{E}]^{-1} \mathbf{F} \mathbf{D}^{-1} & -\mathbf{D}^{-1} \mathbf{E} [\mathbf{G} - \mathbf{F} \mathbf{D}^{-1} \mathbf{E}]^{-1} \\ -[\mathbf{G} - \mathbf{F} \mathbf{D}^{-1} \mathbf{E}]^{-1} \mathbf{F} \mathbf{D}^{-1} & [\mathbf{G} - \mathbf{F} \mathbf{D}^{-1} \mathbf{E}]^{-1} \end{bmatrix} \quad (10-52)$$

(see Appendix IV). Hence, from (10-49), (10-51), and (10-52),

$$\mathcal{H}^{-1} = \begin{bmatrix} \mathbf{A} + \mathbf{R}_c \mathbf{A}^{-T} \mathbf{Q} & -\mathbf{R}_c \mathbf{A}^{-T} \\ -\mathbf{A}^{-T} \mathbf{Q} & \mathbf{A}^{-T} \end{bmatrix} \quad (10-53)$$

where

$$\mathbf{A}^{-T} = (\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1} \quad (10-54)$$

If we denote \mathbf{h} as an eigenvector of \mathcal{H} , and λ as an eigenvalue, then

$$\mathcal{H}\mathbf{h} = \lambda\mathbf{h} \quad (10-55)$$

Let

$$\mathbf{h} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad (10-56)$$

where \mathbf{f} and \mathbf{g} are n -vectors. Substituting (10-49) into (10-55) yields two equations. Also, substituting (10-53) into

$$\mathcal{H}^{-T} \begin{bmatrix} \mathbf{g} \\ -\mathbf{f} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{g} \\ -\mathbf{f} \end{bmatrix} \quad (10-57)$$

yields the same two equations. Thus λ is an eigenvalue of \mathcal{H}^{-T} and hence \mathcal{H}^{-1} , and therefore $1/\lambda$ is an eigenvalue of \mathcal{H} . Thus the eigenvalues of \mathcal{H} are such that the reciprocal of every eigenvalue is also an eigenvalue.

Next we define the vectors $\mathbf{v}(k)$ and $\mathbf{s}(k)$ and the similarity transformation \mathbf{W} (see Section 2.9) such that

$$\begin{bmatrix} \mathbf{x}^o(k) \\ \mathbf{p}^o(k) \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{v}(k) \\ \mathbf{s}(k) \end{bmatrix} \quad (10-58)$$

Then, for the case of distinct eigenvalues of \mathcal{H} , from (10-48) we write

$$\begin{bmatrix} \mathbf{v}(k) \\ \mathbf{s}(k) \end{bmatrix} = \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{v}(k+1) \\ \mathbf{s}(k+1) \end{bmatrix} = \mathbf{W}^{-1} \mathcal{H} \mathbf{W} \begin{bmatrix} \mathbf{v}(k+1) \\ \mathbf{s}(k+1) \end{bmatrix} \quad (10-59)$$

where Λ is a diagonal matrix of the eigenvalues of \mathcal{H} that occur outside the unit circle. Then, in (10-58), the columns of \mathbf{W} are the eigenvectors of \mathcal{H} , and \mathbf{W} is the modal matrix, as shown in Section 2.9.

Next, let the matrix \mathbf{W} be partitioned into four $n \times n$ matrices.

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22} \end{bmatrix} \quad (10-60)$$

Since, from (10-43),

$$\mathbf{p}^o(N) = \mathbf{Q}\mathbf{x}^o(N)$$

from (10-58) and (10-60),

$$\mathbf{p}^o(N) = [\mathbf{W}_{21} \mathbf{v}(N) + \mathbf{W}_{22} \mathbf{s}(N)] = \mathbf{Q}\mathbf{x}^o(N) = \mathbf{Q}[\mathbf{W}_{11} \mathbf{v}(N) + \mathbf{W}_{12} \mathbf{s}(N)] \quad (10-61)$$

We solve this equation for $s(N)$.

$$s(N) = -[W_{22} - QW_{12}]^{-1}[W_{21} - QW_{11}]v(N) = Tv(N) \quad (10-62)$$

which defines the matrix T . Also, for (10-59), we can express $v(N)$ and $s(N)$ as, with j as a positive integer,

$$v(N) = \Lambda^{-j}v(N-j) \quad (10-63)$$

$$s(N) = \Lambda^j s(N-j) \quad (10-64)$$

Thus, from (10-62) and (10-64),

$$s(N-j) = \Lambda^{-j}s(N) = \Lambda^{-j}T\Lambda^{-j}v(N-j) = G(j)v(N-j) \quad (10-65)$$

which defines the matrix $G(j)$.

From (10-58) and (10-60),

$$x^o(N-j) = W_{11}v(N-j) + W_{12}s(N-j) \quad (10-66)$$

$$p^o(N-j) = W_{21}v(N-j) + W_{22}s(N-j) \quad (10-67)$$

Eliminating $v(N-j)$ and $s(N-j)$ for (10-65), (10-66), and (10-67) yields

$$\begin{aligned} p^o(N-j) &= [W_{21} + W_{22}G(j)][W_{11} + W_{12}G(j)]^{-1}x^o(N-j) \\ &= M(j)x^o(N-j) \end{aligned} \quad (10-68)$$

which defines the matrix $M(j)$. This equation is a nonrecursive solution to the state and costate equations (10-4) and (10-44).

To find the optimal gain, $K(k)$, of (10-36), we let $N-j = k+1$ in (10-68).

$$p^o(k+1) = M(N-k-1)x^o(k+1) = M(N-k-1)[Ax^o(k) + Bu^o(k)] \quad (10-69)$$

Since the optimal control from (10-45) is

$$u^o(k) = -R^{-1}B^T p^o(k+1) \quad (10-70)$$

solving these two equations for $u^o(k)$ yields

$$u^o(k) = -[R + B^T M(N-k-1)B]^{-1}B^T M(N-k-1)Ax^o(k)$$

Comparing this equation with (10-37) and (10-38), we see that

$$M(N-k-1) = P(k+1) \quad (10-71)$$

Then, from (10-65) and (10-68),

$$M(k+1) = [W_{21} + W_{22}G(N-k-1)][W_{11} + W_{12}G(N-k-1)]^{-1} \quad (10-72)$$

where

$$G(N-k-1) = -\Lambda^{-(N-k-1)}[W_{22} - QW_{12}]^{-1}[W_{21} - QW_{11}]\Lambda^{-(N-k-1)} \quad (10-73)$$

The optimal gain is given by, from (10-70),

$$K(k) = [R + B^T M(N-k-1)B]^{-1}B^T M(N-k-1)A \quad (10-74)$$

and the nonrecursive solution of LQ design is (10-72) and (10-74).

For the steady-state solution such that the gains in (10-38) have become constant values, it must then be true in (10-47) that

$$\mathbf{P}(k) = \mathbf{P}(k + 1) = \text{constant matrix} \quad (10-75)$$

We will denote this constant matrix as $\hat{\mathbf{P}}$. Then (10-47) becomes

$$\hat{\mathbf{P}} = \mathbf{A}^T \hat{\mathbf{P}} \mathbf{A} + \mathbf{Q} - \mathbf{A}^T \hat{\mathbf{P}} \mathbf{B} [\mathbf{B}^T \hat{\mathbf{P}} \mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \hat{\mathbf{P}} \mathbf{A} \quad (10-76)$$

This equation is referred to as the *algebraic Riccati equation*. Perhaps the simplest approach to finding the solution of this equation is that indicated in Example 10.3—set N to a large value and calculate the values of the \mathbf{P} matrix (by computer) until the matrix elements become constant values. Then we have the solution to (10-76).

We may also find the solution to the algebraic Riccati equation from the foregoing nonrecursive solution to the discrete Riccati equation. In (10-68),

$$\mathbf{p}^o(N - j) = \mathbf{M}(j) \mathbf{x}^o(N - j) \quad (10-77)$$

The steady-state solution is obtained by allowing j to approach ∞ . From (10-65) and (10-68)

$$\lim_{j \rightarrow \infty} \mathbf{M}(j) = \mathbf{W}_{21} \mathbf{W}_{11}^{-1} \quad (10-78)$$

since \mathbf{A} contains the eigenvalues outside the unit circle, and thus

$$\lim_{j \rightarrow \infty} \mathbf{A}^{-j} = \mathbf{0}$$

Hence, from (10-71), the solution to the algebraic Riccati equation (10-76) is

$$\hat{\mathbf{P}} = \mathbf{W}_{21} \mathbf{W}_{11}^{-1} = \lim_{j \rightarrow \infty} \mathbf{M}(j) \quad (10-79)$$

and, for (10-74),

$$\hat{\mathbf{K}} = [\mathbf{R} + \mathbf{B} \hat{\mathbf{P}} \mathbf{B}^T]^{-1} \mathbf{B}^T \hat{\mathbf{P}} \mathbf{A} \quad (10-80)$$

Example 10.4

The nonrecursive method of solution of the optimal control problem will be illustrated using the first-order system of Example 10.1. We wish to design an optimal control law for the system

$$x(k + 1) = 2x(k) + u(k)$$

with the cost function

$$J_N = \sum_{k=0}^N x^2(k) + u^2(k)$$

Thus the required parameters are

$$A = 2 \quad Q = 1$$

$$B = 1 \quad R = 1$$

and

$$R_C = \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T = 1$$

Then in (10-49),

$$\mathcal{K} = \begin{bmatrix} A^{-1} & A^{-1}R_C \\ QA^{-1} & A^T + QA^{-1}R_C \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 2.5 \end{bmatrix}$$

The eigenvalues of \mathcal{K} satisfy the equation

$$|\lambda I - \mathcal{K}| = (\lambda - 0.5)(\lambda - 2.5) - 0.25 = \lambda^2 - 3\lambda + 1 = 0$$

Thus the eigenvalues are 2.618 and 0.382. The eigenvectors satisfy the equation

$$\mathcal{K}\mathbf{h} = \lambda\mathbf{h}$$

Thus the eigenvectors are

$$\mathbf{h}_1 = \begin{bmatrix} 1 \\ 4.237 \end{bmatrix}, \quad \mathbf{h}_2 = \begin{bmatrix} 1 \\ -0.236 \end{bmatrix}$$

and the similarity transformation \mathbf{W} is

$$\mathbf{W} = [\mathbf{h}_1 \quad \mathbf{h}_2] = \begin{bmatrix} 1 & 1 \\ 4.237 & -0.236 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

Then, from (10-59),

$$\begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda^{-1} \end{bmatrix} = \mathbf{W}^{-1}\mathcal{K}\mathbf{W} = \begin{bmatrix} 2.618 & 0 \\ 0 & 0.382 \end{bmatrix}$$

From (10-62),

$$T = -[W_{22} - QW_{12}]^{-1}[W_{21} - QW_{11}] = 2.619$$

and from (10-65),

$$G(j) = \Lambda^{-j}T\Lambda^{-j} = 2.619(0.382)^{2j}$$

From (10-71) and (10-72),

$$P(k+1) = M(N-k-1) = \frac{4.237 - 0.618(0.382)^{2(N-k-1)}}{1 + 2.619(0.382)^{2(N-k-1)}}$$

and the optimal gain, from (10-74), is

$$K(k) = \frac{2M(N-k-1)}{1 + M(N-k-1)}$$

For $N = 2$ and $k = 1$,

$$K(1) = \frac{2M(0)}{1 + M(0)} = \frac{2(1)}{1 + 1} = 1$$

and for $k = 0$,

$$K(0) = \frac{2M(1)}{1 + M(1)} = \frac{2(3)}{1 + 3} = 1.5$$

These values check those of Example 10.1. The steady-state value, \hat{P} , is obtained from (10-79).

$$\hat{P} = W_{21}W_{11}^{-1} = 4.237$$

and the steady-state gain is

$$\hat{K} = \frac{2(4.237)}{1 + 4.237} = 1.618$$

Some general results regarding the steady-state solution of the discrete Riccati equation, (10-47), will now be given. Note that the steady-solution of this equation is the solution with N finite and $m \rightarrow \infty$.

First we define the term *stabilizable*.

Definition. The discrete-time system (10-4) is said to be stabilizable if there exists a matrix K such that the eigenvalues of $A - BK$ are all inside the unit circle.

Then we may state the following theorem.

Theorem 1. If the system (10-4) is either controllable or stabilizable, the discrete Riccati equation (10-47) has a limiting solution as $m \rightarrow \infty$; that is,

$$\lim_{m \rightarrow \infty} P(N - m) = \hat{P}$$

as $k \rightarrow -\infty$. This theorem is given without proof, as is the following theorem [4].

Theorem 2. If the system (10-4) is observable, then \hat{P} is positive definite and the optimal closed-loop system is asymptotically stable.

10.7 LEAST-SQUARES CURVE FITTING

In this and the following two sections we introduce *system identification* by the least-squares procedure. Many different techniques are available for finding a linear model of a physical system by using input-output measurements [5,6]; we will consider only least-squares system identification. Least-squares curve fitting will be used to introduce this topic.

Suppose that we suspect a linear relationship between the variables x and y of the form

$$y = kx \quad (10-81)$$

where k is a constant. We are able to make measurements and obtain data pairs (x_i, y_i) , and wish to calculate the "best" estimate of k from the data. The functional relationship (10-81) can be expressed as, using the data pairs,

$$\begin{aligned} y_1 &= kx_1 + e_1 \\ y_2 &= kx_2 + e_2 \\ y_3 &= kx_3 + e_3 \\ &\vdots \end{aligned} \quad (10-82)$$

In this equation, (x_i, y_i) represent the data obtained from the i th measurement. The terms e_i are the errors, and are present because of errors in taking data, the inexactness of (10-81), and so on. Note that if no errors are present, we can determine k exactly from any one data pair. However, errors are always present in physical situations, and solving for k from different data pairs will generally yield different values of k . We wish to solve for k by a method that minimizes the errors, in some sense.

We can write (10-82) in vector form as

$$\mathbf{y} = k\mathbf{x} + \mathbf{e} \quad (10-83)$$

where, for example, for three data pairs,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (10-84)$$

The case for $N = 3$ is illustrated in Figure 10-5. Note that, for N data pairs, the sum of the squared errors is given by

$$\mathbf{e}^T \mathbf{e} = e_1^2 + e_2^2 + \cdots + e_N^2 = \sum_{k=1}^N e_k^2 \quad (10-85)$$

Hence $\mathbf{e}^T \mathbf{e}$ is a scalar. From (10-83) and (10-85), the sum of the squared errors is given by

$$\mathbf{e}^T \mathbf{e} = [\mathbf{y} - k\mathbf{x}]^T [\mathbf{y} - k\mathbf{x}] = \mathbf{y}^T \mathbf{y} - 2k\mathbf{x}^T \mathbf{y} + k^2 \mathbf{x}^T \mathbf{x} \quad (10-86)$$

since $\mathbf{x}^T \mathbf{y} = \mathbf{y}^T \mathbf{x}$. If we choose k to minimize (10-86), the resulting estimate of k is called the *least-squares estimate*.

We obtain the least-squares estimate of k from

$$\frac{\partial(\mathbf{e}^T \mathbf{e})}{\partial k} = -2\mathbf{x}^T \mathbf{y} + 2k\mathbf{x}^T \mathbf{x} = 0 \quad (10-87)$$

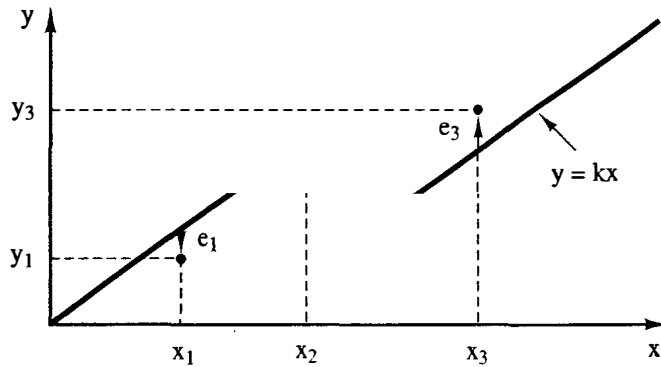


Figure 10-5 Least-squares curve fitting.

Solving this equation for k yields

$$\hat{k} = (\mathbf{x}^T \mathbf{x})^{-1} \mathbf{x}^T \mathbf{y} = \frac{\mathbf{x}^T \mathbf{y}}{\mathbf{x}^T \mathbf{x}} \quad (10-88)$$

where \hat{k} is the least-squares estimate of k (see Problem 10-27). An example will now be given.

Example 10.5

Suppose that $y = kx$ and we wish to determine the least squares estimate of k from the three data pairs

x	y
1.0	1.25
2.1	2.0
2.95	2.9

Then, from (10-88),

$$\hat{k} = \frac{(1.0)(1.25) + (2.1)(2.0) + (2.95)(2.9)}{(1.0)(1.0) + (2.1)(2.1) + (2.95)(2.95)} = \frac{14.005}{14.1125} = 0.9924$$

For example, if x were the input voltage to an amplifier and y the output voltage, the least-squares estimate the gain of the amplifier is 0.9924, for the data given.

10.8 LEAST-SQUARES SYSTEM IDENTIFICATION

In this section *least-squares system identification* is developed. We will see that it is a simple extension of least-squares curve fitting. We assume a system transfer-function model of the form

$$\frac{Y(z)}{U(z)} = G(z) = \frac{b_1 z^{n-1} + b_2 z^{n-2} + \cdots + b_n}{z^n - a_1 z^{n-1} - \cdots - a_n} \quad (10-89)$$

where $U(z)$ is the input and $Y(z)$ is the output. Hence the system is described by the difference equation

$$\begin{aligned} y(k) = & a_1 y(k-1) + a_2 y(k-2) + \cdots + a_n y(k-n) \\ & + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_n u(k-n) \end{aligned} \quad (10-90)$$

This model is often referred to as the ARMA (autoregressive moving-average) model. We wish to determine the coefficient vector

$$\boldsymbol{\theta} = (a_1 \ a_2 \cdots a_n \ b_1 \ b_2 \cdots b_n)^T \quad (10-91)$$

from measurements of the input-output sequences $u(k)$ and $y(k)$.

To illustrate the procedure, we first consider the first-order case, with

$$\frac{Y(z)}{U(z)} = G(z) = \frac{b_1}{z - a_1}$$

Hence

$$y(k) = a_1 y(k-1) + b_1 u(k-1)$$

and thus

$$y(1) = a_1 y(0) + b_1 u(0) + e(1)$$

$$y(2) = a_1 y(1) + b_1 u(1) + e(2)$$

$$y(3) = a_1 y(2) + b_1 u(2) + e(3)$$

where the error terms $e(k)$ occur because of measurement inaccuracies, model inaccuracies, and so on. This equation can be expressed in vector-matrix form as

$$\begin{bmatrix} y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} y(0) & u(0) \\ y(1) & u(1) \\ y(2) & u(2) \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} e(1) \\ e(2) \\ e(3) \end{bmatrix}$$

which may be expressed as

$$\mathbf{y}(3) = \mathbf{F}(3)\boldsymbol{\theta} + \mathbf{e}(3) \quad (10-92)$$

We wish then to calculate the coefficient vector $\boldsymbol{\theta}$ that will minimize the sum of the squared errors. The general procedure will now be developed.

Consider the n th-order ARMA model

$$\begin{aligned} \text{[eq. (10-90)]} \quad y(k) &= a_1 y(k-1) + a_2 y(k-2) + \cdots + a_n y(k-n) \\ &\quad + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_n u(k-n) \end{aligned}$$

with the set of $(N+1)$ measurement pairs.

$$\{u(0), y(0)\}, \{u(1), y(1)\}, \dots, \{u(N), y(N)\} \quad (10-93)$$

with $N \geq n$. Define the vector $\mathbf{f}(k)$ by

$$\mathbf{f}^T(k) = [y(k-1) \ y(k-2) \cdots y(k-n) \ u(k-1) \cdots u(k-n)] \quad (10-94)$$

The first error that is a function of only known data measurements is $e(n)$. Then, for the sample periods $n, n+1, \dots, N$,

$$y(n) = \mathbf{f}^T(n)\boldsymbol{\theta} + e(n)$$

$$y(n+1) = \mathbf{f}^T(n+1)\boldsymbol{\theta} + e(n+1) \quad (10-95)$$

$$\vdots$$

$$y(N) = \mathbf{f}^T(N)\boldsymbol{\theta} + e(N)$$

where θ is given in (10-91). Using the notation

$$y(N) = \begin{bmatrix} y(n) \\ y(n+1) \\ \vdots \\ y(N) \end{bmatrix}, \quad F(N) = \begin{bmatrix} f^T(n) \\ f^T(n+1) \\ \vdots \\ f^T(N) \end{bmatrix}, \quad e(N) = \begin{bmatrix} e(n) \\ e(n+1) \\ \vdots \\ e(N) \end{bmatrix} \quad (10-96)$$

we can express (10-95) as

$$y(N) = F(N)\theta + e(N) \quad (10-97)$$

In (10-96) and (10-97), $y(N)$ is of order $(N - n + 1) \times 1$, $F(N)$ is $(N - n + 1) \times 2n$, θ is $2n \times 1$, and $e(N)$ is $(N - n + 1) \times 1$.

Next the cost function $J(\theta)$ is defined as the sum of the squared errors:

$$J(\theta) = \sum_{k=n}^N e^2(k) = e^T(N)e(N) \quad (10-98)$$

Then, from (10-97) and (10-98),

$$\begin{aligned} J(\theta) &= [y - F\theta]^T[y - F\theta] = y^T y - \theta^T F^T y - y^T F\theta + \theta^T F^T F\theta \\ &= y^T y - 2\theta^T F^T y + \theta^T F^T F\theta \end{aligned} \quad (10-99)$$

where the notational dependence of the terms on N has been omitted for convenience. Thus the value of θ that minimizes $J(\theta)$ satisfies the equation

$$\frac{\partial J(\theta)}{\partial \theta} = -2F^T y + 2F^T F\theta = 0 \quad (10-100)$$

from (10-14) and (10-15). Or,

$$F^T F\theta = F^T y \quad (10-101)$$

The least-squares estimate of θ is then

$$\hat{\theta}_{LS} = [F^T(N)F(N)]^{-1} F^T(N)y(N) \quad (10-102)$$

provided that the indicated inverse matrix exists. If the input sequence $u(k)$ is *persistently exciting* (all system dynamics are sufficiently excited by the input) and if θ is *identifiable* (the parameters can be uniquely determined from the data) [5,6], the inverse matrix will exist.

The structure of (10-102) is somewhat difficult to understand, because of the matrix transposes. As an example of this structure, consider a second-order system with four data points. Then $N = 4$ and

$$F^T(4)y(4) = \begin{bmatrix} y(1) & y(0) & u(1) & u(0) \\ y(2) & y(1) & u(2) & u(1) \\ y(3) & y(2) & u(3) & u(2) \end{bmatrix}^T \begin{bmatrix} y(2) \\ y(3) \\ y(4) \end{bmatrix} \quad (10-103)$$

Note from (10-96) that the total number of data points is $(N + 1)$, and only the output in the last data point is used.

An example will now be given.

Example 10.6

Suppose that a first-order system yields the following data.



k	$u(k)$	$y(k)$
0	1.0	0
1	0.75	0.3
2	0.50	0.225

The assumed transfer function is

$$G(z) = \frac{b_1}{z - a_1}, \quad \theta = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}$$

Thus

$$y(k) = a_1 y(k-1) + b_1 u(k-1) = [y(k-1) \quad u(k-1)] \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \mathbf{f}^T(k) \theta$$

From (10-96), since $N = 2$,

$$\mathbf{F}(2) = \begin{bmatrix} \mathbf{f}^T(1) \\ \mathbf{f}^T(2) \end{bmatrix} = \begin{bmatrix} y(0) & u(0) \\ y(1) & u(1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0.3 & 0.75 \end{bmatrix}$$

Then

$$\mathbf{F}^T \mathbf{F} = \begin{bmatrix} 0 & 0.3 \\ 1 & 0.75 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0.3 & 0.75 \end{bmatrix} = \begin{bmatrix} 0.09 & 0.225 \\ 0.225 & 1.5625 \end{bmatrix}$$

and the inverse of this matrix is calculated to be

$$[\mathbf{F}^T \mathbf{F}]^{-1} = \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix}$$

The least-squares estimate of θ is then, from (10-102),

$$\begin{aligned} \hat{\theta}_{LS} &= [\mathbf{F}^T \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{y} = \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0.3 \\ 1 & 0.75 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.225 \end{bmatrix} \\ &= \begin{bmatrix} -2.5 & 3.333 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.225 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \end{aligned}$$

Thus the transfer function is identified to be

$$G(z) = \frac{0.3}{z}$$

with the difference equation

$$y(k) = 0.3u(k-1)$$

and the measurement data given above is seen to satisfy this difference equation exactly. For this example, the measurement data was initially generated from this difference equation. In a physical situation, the data would not fit exactly, and in fact the errors $e(k)$ may be quite large.

A MATLAB program that performs the calculations of this example is given by

```
format compact
F = [0 1; 0.3 0.75];
Y = [.3; 0.225];
thetals = inv(F'*F)*F'*Y
```

In some cases we would like to weight the errors in the form

$$J(\theta) = w(n)e^2(n) + w(n+1)e^2(n+1) + \cdots + w(N)e^2(N) \quad (10-104)$$

One common case is to weight the most recent data most heavily. In (10-104), the cost function can be expressed as

$$J(\theta) = \sum_{k=n}^N w(k)e^2(k) = \mathbf{e}^T(N)\mathbf{W}(N)\mathbf{e}(N) \quad (10-105)$$

where \mathbf{W} is a diagonal matrix with elements $w_{ii} = w(i+n-1)$. If this cost function is minimized in the manner given above, the resulting weighted least-squares estimate is (see Problem 10-28)

$$\hat{\theta}_{wLS} = [\mathbf{F}^T(N)\mathbf{W}(N)\mathbf{F}(N)]^{-1}\mathbf{F}^T(N)\mathbf{W}(N)\mathbf{y}(N) \quad (10-106)$$

A weighting term that is commonly used is

$$w(k) = a\gamma^{N-k}, \quad \gamma \leq 1 \quad (10-107)$$

If a is chosen such that $a = (1 - \gamma)$, the weighting is said to be *exponential*. For γ small compared to unity, the most recent data dominates the estimation. As γ approaches unity, the more distant data has a larger influence. In (10-107), if $a = \gamma = 1$, the weighting matrix $\mathbf{W}(N) = \mathbf{I}$, and (10-106) becomes the equation for ordinary least-squares identification, (10-102).

10.9 RECURSIVE LEAST-SQUARES SYSTEM IDENTIFICATION

If all measurements are accumulated and the coefficient vector calculated in (10-106), the process is said to be *batch*. However, (10-106) can be manipulated into a difference equation such that the θ vector can be recalculated with each set of data as it arrives; the solution is then said to be *recursive*. In this section we derive the recursive least-squares identification equations.

From (10-94) and (10-95),

$$\begin{aligned} y(k) &= \mathbf{f}^T(k)\theta + e(k) \\ &= [y(k-1) \ y(k-2) \cdots u(k-1) \cdots]\theta + e(k) \end{aligned}$$

Now, from (10-96),

$$\mathbf{F}^T(N+1) = [\mathbf{f}(n) \cdots \mathbf{f}(N)\mathbf{f}(N+1)] \quad (10-108)$$

We will let the weighting factor be

$$w(k) = a\gamma^{N+1-k}, \quad \gamma \leq 1 \quad (10-109)$$

In (10-106) we manipulate the term

$$\begin{aligned} \mathbf{F}^T(N+1)\mathbf{W}(N+1)\mathbf{F}(N+1) &= \sum_{k=n}^{N+1} \mathbf{f}(k)a\gamma^{N+1-k}\mathbf{f}^T(k) \\ &= \sum_{k=n}^N \mathbf{f}(k)a\gamma\gamma^{N-k}\mathbf{f}^T(k) + \mathbf{f}(N+1)a\mathbf{f}^T(N+1) \\ &= \gamma\mathbf{F}^T(N)\mathbf{W}(N)\mathbf{F}(N) + \mathbf{f}(N+1)a\mathbf{f}^T(N+1) \end{aligned} \quad (10-110)$$

This relatively simple form results from $\mathbf{W}(N)$ being diagonal.

For convenience the $2n \times 2n$ matrix $\mathbf{P}(k)$ is defined as

$$\mathbf{P}(k) = [\mathbf{F}^T(k)\mathbf{W}(k)\mathbf{F}(k)]^{-1} \quad (10-111)$$

Then, from (10-110) and (10-111),

$$\mathbf{P}^{-1}(N+1) = \gamma\mathbf{P}^{-1}(N) + \mathbf{f}(N+1)a\mathbf{f}^T(N+1) \quad (10-112)$$

The matrix inversion lemma is given by (see Appendix IV)

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1} \quad (10-113)$$

We let the right side of (10-112) be $(\mathbf{A} + \mathbf{BCD})$ with the following assignments:

$$\begin{aligned} \mathbf{A} &= \gamma\mathbf{P}^{-1}(N) & \mathbf{B} &= \mathbf{f}(N+1) \\ \mathbf{C} &= a & \mathbf{D} &= \mathbf{f}^T(N+1) \end{aligned} \quad (10-114)$$

The matrix inverse of (10-112) can then be expressed as

$$\begin{aligned} \mathbf{P}(N+1) &= \frac{1}{\gamma}\mathbf{P}(N) - \frac{1}{\gamma}\mathbf{P}(N)\mathbf{f}(N+1) \\ &\quad \times \left[\frac{1}{a} + \mathbf{f}^T(N+1)\frac{1}{\gamma}\mathbf{P}(N)\mathbf{f}(N+1) \right]^{-1} \mathbf{f}^T(N+1)\frac{1}{\gamma}\mathbf{P}(N) \end{aligned} \quad (10-115)$$

Since this equation is valid for any $N \geq n$, we have developed a difference equation for $\mathbf{P}(k)$. Note that the factor in the inverse in (10-115) is scalar, and thus no matrix inversion is required.

The second factor in the batch equation for $\boldsymbol{\theta}$, (10-106), can be manipulated as

$$\begin{aligned} \mathbf{F}^T\mathbf{W}\mathbf{y} &= [\mathbf{f}(n) \dots \mathbf{f}(N+1)] \begin{bmatrix} a\gamma^{N+1-n} & & \\ & \ddots & \\ & & a\gamma \\ & & & a \end{bmatrix} \begin{bmatrix} y(n) \\ \vdots \\ y(N) \\ y(N+1) \end{bmatrix} \\ &= \gamma\mathbf{F}^T(N)\mathbf{W}(N)\mathbf{y}(N) + \mathbf{f}(N+1)a\mathbf{y}(N+1) \end{aligned} \quad (10-116)$$

where each matrix of the left side is a function of $(N+1)$, and only the nonzero elements of $\mathbf{W}(N+1)$ are shown.

The substitution of (10-111), (10-115), and (10-116) into the weighted least-squares solution (10-106) yields the following set of difference equations, after some manipulations (see Problem 10-29).

$$\mathbf{P}(N) = [\mathbf{F}^T(N)\mathbf{W}(N)\mathbf{F}(N)]^{-1} \quad (10-117)$$

$$\mathbf{L}(N+1) = \frac{1}{\gamma} \mathbf{P}(N) \mathbf{f}(N+1) \left[\frac{1}{a} + \mathbf{f}^T(N+1) \frac{1}{\gamma} \mathbf{P}(N) \mathbf{f}(N+1) \right]^{-1} \quad (10-118)$$

$$\hat{\boldsymbol{\theta}}_{wLS}(N+1) = \hat{\boldsymbol{\theta}}_{wLS}(N) + \mathbf{L}(N+1)[y(N+1) - \mathbf{f}^T(N+1)\hat{\boldsymbol{\theta}}_{wLS}(N)] \quad (10-119)$$

$$\mathbf{P}(N+1) = \frac{1}{\gamma} [\mathbf{I} - \mathbf{L}(N+1)\mathbf{f}^T(N+1)]\mathbf{P}(N) \quad (10-120)$$

These equations apply for $N \geq n$; hence, if N is replaced with k , the difference equations form the recursive solution for the weighted least-squares system identification. The solution is started by finding $\mathbf{P}(k)$ and $\hat{\boldsymbol{\theta}}(k)$ for some value of k . Next the vector $\mathbf{f}(k+1)$ is formed, where

$$\mathbf{f}(k+1) = [y(k) \quad y(k-1) \cdots y(k-n) \quad u(k) \cdots u(k-n)]^T$$

The next set of measurements are used to form $y(k+1)$, and equations (10-118), (10-119), and (10-120) are then solved recursively. The initial solution for $\mathbf{P}(k)$ from (10-117) requires a matrix inversion. For that reason, the initial values of $\mathbf{P}(k)$ and $\hat{\boldsymbol{\theta}}(k)$ are estimated (not calculated) in some computer programs, and several iterations are required before the estimated coefficient vector settles to approximately constant values.

Recall that the least-squares identification gives a “best” fit to the data. However, the “best” fit may be so inaccurate as to be useless. For example, suppose that for a given physical system, accurate modeling requires a second-order linear model. If we try to fit the data to a first-order model, the least-squares procedure will yield a “best” first-order transfer function. However, the transfer function will not accurately model the physical system. The conclusion is that additional verification is required, once a transfer function is calculated. One such verification is the comparison of the time responses of the physical system and the derived model. Also, the frequency response of the physical system can be measured and compared with the frequency response of the derived model.

An additional problem with any least-squares procedure is that if one data pair is greatly in error, the square of this large error will tend to dominate the sum-of-squared errors. Thus the procedure will be heavily weighted to reduce this large error, resulting in a less accurate model. For this reason it is generally recommended that any data pairs that obviously do not match the remaining data be ignored in some consistent manner. An example of recursive least-squares identification will now be given.

Example 10.7

The data of Example 10.6 will be used to illustrate the ordinary least-squares recursive procedure. No weighting of data is employed, and $a = \gamma = 1$ in (10-117)–(10-120).

Suppose that an additional data point is taken, with $y(3) = 0.15, u(3) = 0.40$. We use the solution in Example 10.6 to start the recursive solution; hence $n = 1$ and $N = 2$. Thus, from (10-117) and Example 10.6,

$$\mathbf{P}(N) = \mathbf{P}(2) = [\mathbf{F}^T(2)\mathbf{F}(2)]^{-1} = \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix}$$

and

$$\hat{\boldsymbol{\theta}}(N) = \hat{\boldsymbol{\theta}}(2) = \begin{bmatrix} 0 \\ 0.3 \end{bmatrix}$$

From the data of Example 10.6,

$$\mathbf{f}^T(N+1) = \mathbf{f}^T(3) = [y(2) \quad u(2)] = [0.225 \quad 0.5]$$

Then, in (10-118),

$$\begin{aligned} \mathbf{f}^T(3)\mathbf{P}(2)\mathbf{f}(3) &= [0.225 \quad 0.5] \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix} \begin{bmatrix} 0.225 \\ 0.5 \end{bmatrix} \\ &= [2.656 \quad -0.0625] \begin{bmatrix} 0.225 \\ 0.5 \end{bmatrix} = 0.5664 \end{aligned}$$

and

$$\begin{aligned} \mathbf{L}(3) &= \mathbf{P}(2)\mathbf{f}(3)[1 + \mathbf{f}^T(3)\mathbf{P}(2)\mathbf{f}(3)]^{-1} \\ &= \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix} \begin{bmatrix} 0.225 \\ 0.5 \end{bmatrix} [1/0.5664] = \begin{bmatrix} 4.690 \\ -0.110 \end{bmatrix} \end{aligned}$$

From (10-119), the estimated coefficient vector, after four data points, is then

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{WLS}(3) &= \hat{\boldsymbol{\theta}}_{WLS}(2) + \mathbf{L}(3)[y(3) - \mathbf{f}^T(3)\hat{\boldsymbol{\theta}}_{WLS}(2)] \\ &= \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} + \begin{bmatrix} 4.690 \\ -0.110 \end{bmatrix} \left[0.15 - [0.225 \quad 0.5] \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} \right] \\ &= \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} \end{aligned}$$

Note that since the data is exact in this case, no correction in the coefficient vector results.

10.10 OPTIMAL STATE ESTIMATION—KALMAN FILTERS

In this section we extend the state-estimation concepts introduced in Chapter 9. We consider an optimal state estimation technique called Kalman filtering [7,8]. The Kalman filter presented in this section has the equations of the current observer of Section 9.5; the only difference is that the gain matrix \mathbf{G} is calculated by a different procedure. There is also a Kalman filter in the form of the prediction observer [9]; we will not consider that form. In the following, the reader is assumed to have some understanding of random variables and stochastic processes [7,8,10].

The plant is assumed to be described by the equations

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_1\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k)\end{aligned}\quad (10-121)$$

where $\mathbf{x}(k)$ [$n \times 1$] are the states, $\mathbf{u}(k)$ [$r \times 1$] are the known inputs, $\mathbf{w}(k)$ [$s \times 1$] are the random plant disturbances, $\mathbf{y}(k)$ [$p \times 1$] are the measurements, and $\mathbf{v}(k)$ [$p \times 1$] are the random inaccuracies in the measurements. The random inputs $\mathbf{w}(k)$ and $\mathbf{v}(k)$ in (10-121) are assumed to be uncorrelated and to have Gaussian distributions with the properties:

$$\begin{aligned}E[\mathbf{w}(k)] &= \mathbf{0}, \quad E[\mathbf{v}(k)] = \mathbf{0} \\ \text{cov}[\mathbf{w}(j), \mathbf{w}(k)] &= E[\mathbf{w}(j)\mathbf{w}^T(k)] = \mathbf{R}_w \delta_{jk} \\ \text{cov}[\mathbf{v}(j), \mathbf{v}(k)] &= E[\mathbf{v}(j)\mathbf{v}^T(k)] = \mathbf{R}_v \delta_{jk}\end{aligned}\quad (10-122)$$

$E[\cdot]$ is the mathematical expectation operation, and $\text{cov}[\cdot]$ denotes the covariance. In (10-122), δ_{jk} is the Kronecker delta function, defined by

$$\delta_{jk} = \begin{cases} 0, & j \neq k \\ 1, & j = k \end{cases}$$

Hence $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are discrete white-noise sequences with Gaussian distributions.

We will not derive the equations for the Kalman filter; the derivation requires an extensive background in stochastic processes, which is beyond the scope of this book. However, the background acquired in study of observers in Chapter 9 will help to understand the purpose and the properties of the Kalman filter. The mathematical expectation in (10-122) can be considered to be an averaging operation. Hence the expected value of a vector or a matrix can be viewed as the vector or matrix made up of the average values of the elements.

The estimates of the states $\mathbf{x}(k)$ are denoted as $\mathbf{q}(k)$, and the errors of estimation are

$$\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k) \quad (10-123)$$

The covariance of the error vector is denoted as $\mathbf{P}(k)$ [$n \times n$], with

$$\mathbf{P}(k) = E[\mathbf{e}(k)\mathbf{e}^T(k)] \quad (10-124)$$

Hence the diagonal elements of $\mathbf{P}(k)$ are the average squared errors of the estimation. The cost function for the minimization process is chosen as the trace of $\mathbf{P}(k)$ (the sum of the diagonal elements or the sum of the average squared errors of estimation).

$$\begin{aligned}J(k) &= \text{tr } \mathbf{P}(k) = E[e_1^2(k)] + E[e_2^2(k)] + \cdots + E[e_n^2(k)] \\ &= \sigma_{e1}^2(k) + \sigma_{e2}^2(k) + \cdots + \sigma_{en}^2(k)\end{aligned}\quad (10-125)$$

where $\sigma_{ei}^2(k)$ is the variance of $e_i(k)$. This cost function can also be expressed as

$$J(k) = E[\mathbf{e}^T(k)\mathbf{e}(k)]$$

Note the *expected values* of the squared estimation errors are minimized, and not of the actual errors themselves. Since the system (10-121) has random inputs and the measurements have random errors, the actual estimation errors can never be determined. Hence we are forced to consider only the statistical characteristics of the estimation errors, and minimize a function of the expected values (average values) of these errors. Hence the Kalman filter is optimal only on the average.

An additional property of the Kalman filter is that the cost function

$$J'(k) = \text{tr } E[\mathbf{e}(k)\mathbf{Q}\mathbf{e}^T(k)] \quad (10-126)$$

is also minimized, for \mathbf{Q} any positive semidefinite matrix [7]. Hence, for example, for a second-order system, the cost functions

$$J'_1(k) = E[e_1^2(k)]$$

$$J'_2(k) = E[e_1^2(k)] + E[e_2^2(k)]$$

$$J'_3(k) = 10E[e_1^2(k)] + E[e_2^2(k)]$$

are all minimized. In addition, the sum

$$J_N = \sum_{k=0}^N E[e_1^2(k)] + E[e_2^2(k)] = \sum_{k=0}^N E[\mathbf{e}^T(k)\mathbf{e}(k)]$$

is minimized, since the individual terms are minimized. Hence we see that the Kalman filter is optimal in many different ways.

As stated above, the mathematical development required to minimize (10-125) [or (10-126)] will not be given here; however, the resulting Kalman filter equations are [7,8]

$$\mathbf{q}(k) = \bar{\mathbf{q}}(k) + \mathbf{G}(k)[\mathbf{y}(k) - \mathbf{C}\bar{\mathbf{q}}(k)] \quad (10-127)$$

$$\bar{\mathbf{q}}(k+1) = \mathbf{A}\mathbf{q}(k) + \mathbf{B}\mathbf{u}(k)$$

In these equations, $\bar{\mathbf{q}}(k)$ is the predicted state estimate at the sampling instant k , and $\mathbf{q}(k)$ is the actual state estimate at k . The gain matrix $\mathbf{G}(k)$ is the Kalman gain, and is calculated from the covariance equations

$$\mathbf{G}(k) = \mathbf{M}(k)\mathbf{C}^T[\mathbf{C}\mathbf{M}(k)\mathbf{C}^T + \mathbf{R}_v]^{-1}$$

$$\mathbf{P}(k) = \mathbf{M}(k) - \mathbf{G}(k)\mathbf{C}\mathbf{M}(k) \quad (10-128)$$

$$\mathbf{M}(k+1) = \mathbf{A}\mathbf{P}(k)\mathbf{A}^T + \mathbf{B}_1\mathbf{R}_w\mathbf{B}_1^T$$

In these equations, $\mathbf{M}(k)$ is the covariance of the prediction errors:

$$\mathbf{M}(k) = E\{[\mathbf{x}(k) - \bar{\mathbf{q}}(k)][\mathbf{x}(k) - \bar{\mathbf{q}}(k)]^T\} \quad (10-129)$$

$\mathbf{P}(k)$ is defined in (10-124). The second equation in (10-128) can be substituted into the third equation in (10-128), and the result can be expressed as a single difference equation in $\mathbf{M}(k)$.

$$\mathbf{M}(k+1) = \mathbf{A}[\mathbf{I} - \mathbf{G}(k)\mathbf{C}]\mathbf{M}(k)\mathbf{A}^T + \mathbf{B}_1\mathbf{R}_w\mathbf{B}_1^T \quad (10-130)$$

The Kalman gains $G(k)$ can be precalculated, since they are independent of the measurements. Of course, the gains may also be calculated in real time. In either case, it is necessary to estimate both $\bar{q}(0)$, the initial predicted state, and $M(0)$, the covariance of the errors in $\bar{q}(0)$. Once $M(0)$ is estimated, the Kalman gains $G(k)$ can be calculated from (10-128), using R_w and R_v , from (10-122). The Kalman filter equations (10-127) can then be solved in real time, using the gains $G(k)$, the initial estimate $\bar{q}(0)$, and the measurements $y(k)$.

In the filter equations (10-127), the predicted estimate $\bar{q}(k)$ can be eliminated, resulting in the filter equation

$$q(k) = [A - G(k)CA]q(k-1) + [B - G(k)CB]u(k-1) + G(k)y(k) \quad (10-131)$$

For the steady-state Kalman filter, $G(k)$ becomes constant. For this case, the filter equations of (10-127) [or (10-131)] have the characteristic equation

$$|zI - (A - GCA)| = 0 \quad (10-132)$$

If the gain G is determined by some procedure other than (10-128), the filter is called the *current estimator*, as discussed in Section 9.6.

Suppose that the control system is designed by the linear-quadratic optimal procedure of the first part of this chapter, and the steady-state gains K are used. Suppose that, in addition, the steady-state Kalman filter is employed to estimate the states. This design is called the infinite-horizon linear-quadratic-Gaussian (IH-LQG) design. The *implementation* is identical to that for the pole-placement current-observer design of Chapter 9. Hence the control-estimator transfer function, $D_{ce}(z)$, is given by (see Figure 9-8)

$$[\text{eq. (9-73)}] \quad D_{ce}(z) = zK[zI - A + GCA + BK - GCBK]^{-1}G$$

Implementation of full-state feedback control systems using Kalman filters as state estimators can also result in systems that are not robust, as discussed in Section 9-5 [11–13]. Hence the stability margins at the plant input must be checked, to ensure adequate relative stability.

Example 10.8



The plant of Example 10.2, a servomotor, will be used to illustrate a Kalman filter design. This plant was also used in several examples for pole-placement and observer design in Chapter 9. The plant equations are

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} x(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w(k) \\ y(k) &= [1 \ 0]x(k) + v(k) \end{aligned}$$

Recall that the two plant states are motor shaft position and shaft velocity. We will assume that the units of the states are degrees of rotation and degrees/second, respectively. The plant disturbance, $w(k)$, affects both states equally. We will assume that the input covariances in (10-122) are

$$\begin{aligned} E[w(j)w(k)] &= R_w \delta_{jk}, & R_w &= 1 = \sigma_w^2 \\ E[v(j)v(k)] &= R_v \delta_{jk}, & R_v &= 1 = \sigma_v^2 \end{aligned}$$

where σ denotes the standard deviation. For a Gaussian-distributed random variable a with zero mean and variance of σ^2 [8],

$$\text{prob}[|a| \leq \sigma] = 0.683$$

$$\text{prob}[|a| \leq 2\sigma] = 0.955$$

$$\text{prob}[|a| \leq 3\sigma] = 0.997$$

For example, for the measurement inaccuracies, we are saying that 68 percent of the measurements are accurate to within 1° of shaft rotation, and in only 3 of 1000 measurements will the error be greater than 3° , on the average. Hence the probability of a Gaussian random variable with a value within the three-sigma (3σ) value of its mean is almost a certainty. Also, the same numbers apply to the plant disturbance.

Because of the complexity of the equations, the Kalman gains for this example were calculated by computer, and the results are plotted in Figure 10-6. The initial value of the covariance matrix, $\mathbf{M}(0)$, was assumed to be the identity matrix. The gains attain the steady-state values in approximately 15 iterations, or in 1.5 s. The steady-state gains are

$$\mathbf{G} = \begin{bmatrix} 0.636 \\ 0.570 \end{bmatrix}$$

and the Kalman filter equations in (10-127) are completely specified. These equations can now be solved, once the initial estimate of the states, $\bar{\mathbf{q}}(0)$, is made. The steady-state error covariance matrix is calculated to be

$$\mathbf{P} = \begin{bmatrix} 0.636 & 0.570 \\ 0.570 & 0.595 \end{bmatrix}$$

Hence for the steady-state Kalman filter, the standard deviation in the estimation of $x_1(k)$ is $\sigma_{e1} = (0.636)^{1/2} = 0.797^\circ$ and $\sigma_{e2} = 0.771^\circ \text{ s}^{-1}$. For the estimate of the motor shaft position, the error is almost certain to be within the three-sigma value of approximately 2.4° . For example, if the measurement is 7° , the true position of the motor shaft is almost certain to be between 4.6° and 9.4° . If this error is too large, either more accurate instrumentation must be used, or the disturbances on the plant must be reduced. Recall that the filter itself is "best" for the error criteria used.

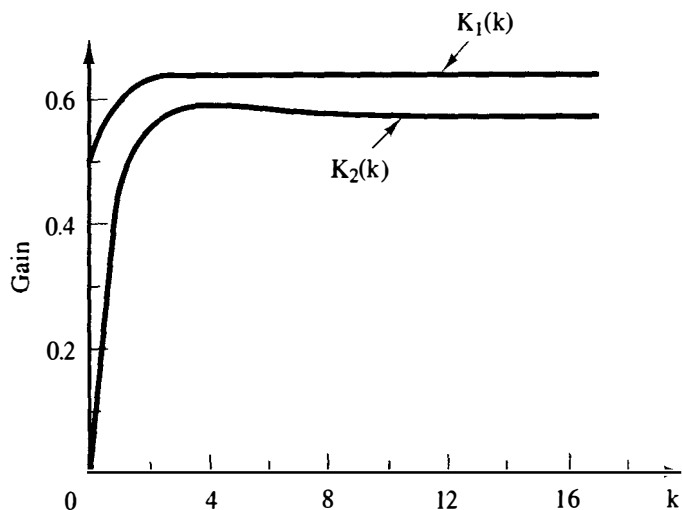


Figure 10-6 Kalman gain versus sample period.

A MATLAB program that calculates the Kalman gains is given by

```
format short e
A = [1 0.0952; 0 0.905];
B1 = [1; 1];
C = [1 0];
Rw = 1;
Rv = 1;
M = [1 0; 0 1];
N = 15;
disp('    k    Gains')
for k=1:N
    G = M*C'*inv(C*M*C' + Rv);
    P = M - G*C*M;
    M = A*P*A' + B1*Rw*B1';
    [k,G']
end
disp(' The final value of the P matrix is:')
P
```

Example 10.9

The design of Example 10.8 will be extended to a steady-state Kalman filter, steady-state LQ (IH-LQG) control system. For the IH-LQG design, the optimal control design of Example 10.3 is used, where

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = 1$$

The steady-state gains from this example are

$$\mathbf{K} = [0.964 \quad 0.719]$$

Hence the IH-LQG controller can be represented by the transfer function (9-73), and evaluation of this transfer function yields

$$D_{ce}(z) = \frac{1.023z^2 - 0.913z}{z^2 - 1.146z + 0.306}$$

The plant transfer function is $G(z)$, and the system open-loop function, for the system opened at the plant input, is

$$D_{ce}(z)G(z) = \frac{1.023z^2 - 0.913z}{z^2 - 1.146z + 0.306} \left(\frac{0.00484z + 0.00468}{z^2 - 1.905z + 0.905} \right)$$

(See Figure 9-8.) Calculation of the frequency response of this function yields a phase margin of 70° and a gain margin of 24 dB. Hence the system has adequate stability margins. It is *always* necessary to check stability margins, since nothing in the design assures adequate margins.

It is informative to consider the IH-LQG design as depicted in Figure 10-7. In this figure all signals shown are vectors; hence the order of multiplication is impor-

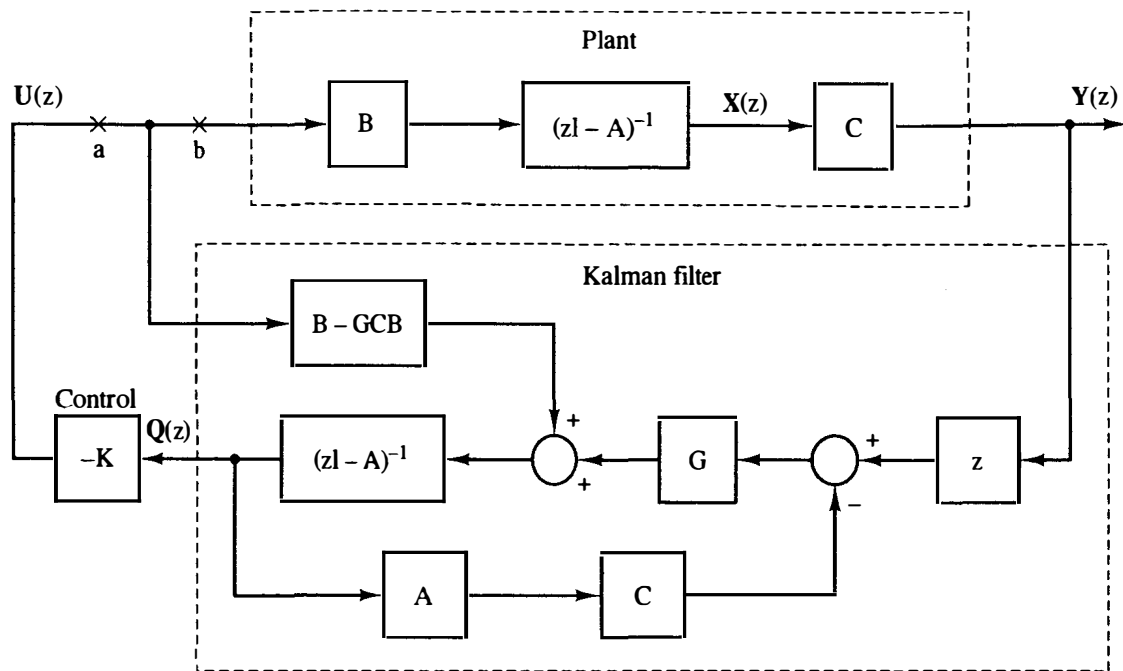


Figure 10-7 IH-LQG control system.

tant. Any simplification on the block diagram must be performed with care. This block diagram can be derived from (10-127), if k is replaced with $(k + 1)$. First the second equation of (10-127) is substituted in the first one.

$$\mathbf{q}(k + 1) = \mathbf{A}\mathbf{q}(k) + [\mathbf{B} - \mathbf{GCB}]\mathbf{u}(k) + \mathbf{G}\mathbf{y}(k + 1) - \mathbf{GCA}\mathbf{q}(k)$$

Taking the z -transform of this equation and rearranging yields

$$(z\mathbf{I} - \mathbf{A})\mathbf{Q}(z) = [\mathbf{B} - \mathbf{GCB}]\mathbf{U}(z) + z\mathbf{G}\mathbf{Y}(z) - \mathbf{GCA}\mathbf{Q}(z)$$

Hence

$$\mathbf{Q}(z) = (z\mathbf{I} - \mathbf{A})^{-1}\{[\mathbf{B} - \mathbf{GCB}]\mathbf{U}(z) + z\mathbf{G}\mathbf{Y}(z) - \mathbf{GCA}\mathbf{Q}(z)\}$$

This is seen to be the equation for $\mathbf{Q}(z)$ at the output of the Kalman filter in Figure 10-7.

The IH-LQG control system as represented in Figure 10-7 is important in understanding the effects of the Kalman filter on the control system. It is seen that the Kalman filter has two inputs. One input is the plant input $\mathbf{U}(z)$ and the other is the plant output $\mathbf{Y}(z)$. If the measurements are specified as being very noisy [the effects of the measurement noise on the estimation is much greater than the effects of the plant disturbances], the resulting value of \mathbf{G} will reduce the effects of the measurements on the state estimates, relative to the effects of the plant input and the plant dynamics. For example, if there are no plant disturbances ($\mathbf{R}_w = \mathbf{0}$), the steady-state Kalman filter will completely ignore the measurements ($\mathbf{G} = \mathbf{0}$) [7]. Of course, this is not desirable under any conditions.

Conversely, if the disturbances on the plant are large, the resulting value of

G will increase the effects of the measurements on the state estimates, and $U(z)$ and the plant dynamics will have less effect. In some cases R_w is purposely increased to indicate model uncertainty.

It is shown in Ref. 11 that the stability margins for the system in Figure 10-7 when opened at point a are always satisfactory. However, the stability margins for the system opened at point b may be quite small. Since the plant model is always suspect, we require large stability margins at point b .

A procedure is given in Ref. 11 for increasing the stability margins at point b , if these margins are small. This is accomplished by increasing the covariance matrix R_w relative to its specified value; however, this increase must be performed in a prescribed manner. The stability margins can be increased, but the resulting Kalman filter is no longer optimal for the original specifications. Hence it is necessary to trade off optimality for stability. Trade-offs of this type are not unusual in applying any theory to physical systems.

10.11 LEAST-SQUARES MINIMIZATION

The three optimal procedures presented in this chapter are all based on least-squares minimization, and hence are related. To show the similarities of the procedures, the equations are listed in Table 10-1 for LQ design, least-squares system identification, and Kalman-filter design. The similarities are evident.

TABLE 10-1 LEAST-SQUARES DESIGN EQUATIONS

<i>Cost Functions</i>	
LQ:	$J_N = \sum_{k=0}^N [\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k)]$
Sys. Id.:	$J = \sum_{k=n}^N e^2(k) = \mathbf{e}^T(k)\mathbf{e}(k)$
Kalman:	$J(k) = \sum_{i=0}^n E[e_i^2(k)] = E[\mathbf{e}^T(k)\mathbf{e}(k)]$
<i>Gain Equations</i>	
LQ:	$\mathbf{K}(k) = [\mathbf{R} + \mathbf{B}^T\mathbf{P}(k+1)\mathbf{B}]^{-1}\mathbf{B}^T\mathbf{P}(k+1)\mathbf{A}$
Sys. Id.:	$\mathbf{L}(k+1) = \mathbf{P}(k)\mathbf{f}(k+1)[1 + \mathbf{f}^T(k+1)\mathbf{P}(k)\mathbf{f}(k+1)]^{-1}$
Kalman:	$\mathbf{G}(k) = \mathbf{M}(k)\mathbf{C}^T[\mathbf{R}_v + \mathbf{C}\mathbf{M}(k)\mathbf{C}^T]^{-1}$
<i>Update Equations</i>	
LQ:	$\mathbf{P}(k) = \mathbf{A}^T\mathbf{P}(k+1)[\mathbf{A} - \mathbf{B}\mathbf{K}(k)] + \mathbf{Q}$
Sys. Id.:	$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{L}(k+1)\mathbf{f}^T(k+1)]\mathbf{P}(k)$
Kalman:	$\mathbf{M}(k+1) = \mathbf{A}[\mathbf{I} - \mathbf{G}(k)\mathbf{C}]\mathbf{M}(k)\mathbf{A}^T + \mathbf{B}_1\mathbf{R}_w\mathbf{B}_1^T$

10.12 SUMMARY

Presented in this chapter were some basic results in linear quadratic optimal control design. Once the cost function has been chosen, the design involves the straightforward solution of a difference equation. Even though the basic formulation is for a finite-time problem, the design procedure is easily extended to the infinite-time problem. The design implementation requires full-state feedback, with the feedback gains time varying for the finite-time problem and constant for the infinite-time problem, provided that the plant is time invariant.

Next least-squares system identification was developed. Modern control design generally requires accurate plant models. Least-squares system identification is an optimal procedure, and uses input-output measurements to calculate the system model. The identification is well suited to computer implementation.

As the final topic, optimal state estimation by Kalman filtering was presented. This filter is a form of the current observer developed in Chapter 9; the only difference is the procedure for calculating the gain matrix G . The infinite-horizon linear-quadratic-Gaussian (IH-LQG) design was then described.

REFERENCES AND FURTHER READING

1. P.M. DeRusso, R. J. Roy, and C. M. Close, *State Variables for Engineers*. New York: John Wiley & Sons, Inc., 1965.
2. R. Bellman, *Adaptive Control Process: A Guided Tour*. Princeton, NJ: Princeton University Press, 1961.
3. *IEEE Trans. Autom. Control*, Bellman Special Issue, Vol. AC-26, Oct. 1981.
4. P. Dorato and A. H. Levis, "Optimal Linear Regulators: The Discrete Time Case," *IEEE Trans. Autom. Control*, Vol. AC-16, pp. 613-620, Dec. 1971.
5. L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*. Cambridge, MA: The MIT Press, 1983.
6. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1980.
7. P. S. Maybeck, *Stochastic Models, Estimation, and Control*, Vol. 1. Orlando, FL: Academic Press, Inc., 1979.
8. R. G. Brown, *Introduction to Random Signal Analysis and Kalman Filtering*. New York: John Wiley & Sons, Inc., 1983.
9. B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall, 1979.
10. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed. New York: McGraw-Hill Book Company, 1984.
11. J. C. Doyle and G. Stein, "Robustness with Observers," *IEEE Trans. Automat. Control*, Vol. AC-24, pp. 607-611, Aug. 1979.
12. B. T. Oranc, "A Classical Approach to Robust Design of Linear-Quadratic-Gaussian Controllers," Ph.D. dissertation, Auburn University, Auburn, AL, 1987.

13. B. E. Sturgis, Jr., "A Study of Stability Margin Problems in Systems Implementing Kalman Filters," M.S. thesis, Auburn University, Auburn, AL, 1984.
14. T. E. Fortman, "A Matrix Inversion Identity," *IEEE Trans. Autom. Control*, Vol. AC-15, p. 599, Oct. 1970.
15. R. Gran and F. Kozin, *Applied Digital Control Systems*. George Washington University Short Course Notes, Washington, D.C., Aug. 1979.
16. D. R. Vaughan, "A Nonrecursive Algebraic Solution for the Discrete Riccati Equation," *IEEE Trans. Autom. Control*, Vol. AC-15, pp. 597-599, Oct. 1970.

PROBLEMS

- 10-1. Given that (10-21) and (10-22) are valid, derive (10-23).
- 10-2. Show that (10-35) can also be expressed as

$$\mathbf{P}(N-m) = \mathbf{A}^T [\mathbf{P}(N-m+1) - \mathbf{P}(N-m+1) \mathbf{B} \mathbf{D} \mathbf{B}^T \mathbf{P}(N-m+1)] \mathbf{A} + \mathbf{Q}$$

where $\mathbf{D} = [\mathbf{B}^T \mathbf{P}(N-m+1) \mathbf{B} + \mathbf{R}]^{-1}$.

- 10-3. Given the discrete system

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

with the cost function

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k)$$

show that the optimal gains which minimize J_N are unchanged if the elements both in \mathbf{Q} and in \mathbf{R} are multiplied by the positive scalar β .

- 10-4. It is shown in Ref. 6 that given the partitioned matrix

$$\mathcal{H} = \begin{bmatrix} \mathbf{D} & \mathbf{E} \\ \mathbf{F} & \mathbf{G} \end{bmatrix}$$

where each partition is $n \times n$, the determinant of \mathcal{H} is given by

$$|\mathcal{H}| = |\mathbf{G}| |\mathbf{D} - \mathbf{E} \mathbf{G}^{-1} \mathbf{F}| = |\mathbf{D}| |\mathbf{G} - \mathbf{F} \mathbf{D}^{-1} \mathbf{E}|$$

Show that the determinant of \mathcal{H} in (10-49) is equal to unity. Considering the development in Section 10.6, is this result expected?

- 10-5. Give a first-order time-invariant discrete system with a cost function

$$J_N = \sum_{k=0}^N Qx^2(k) + Ru^2(k)$$

Show that the optimal gains are a function of only the ratio

$$\alpha = \frac{Q}{R}$$

and not of Q and R singly.

- 10-6. Given the first-order plant described by

$$x(k+1) = 0.9x(k) + 0.1u(k)$$

with the cost function

$$J_3 = \sum_{k=0}^3 [x^2(k) + 5u^2(k)]$$

- (a) Calculate the feedback gains required to minimize the cost function, using the partial-differentiation procedure of Section 10.3.
- (b) Repeat part (a) using the difference-equation approach of Section 10.4.
- (c) Find the maximum magnitude of $u(k)$ as a function of $x(0)$.
- (d) Verify all calculations by computer.

10-7. Given the plant of Problem 10-6, with the cost function

$$J_3 = \sum_{k=0}^3 x^2(k)$$

- (a) Calculate the feedback gains required to minimize the cost function, using the difference-equation approach of Section 10.4.
- (b) Find the maximum magnitude of $u(k)$ as a function of $x(0)$.
- (c) Compare the maximum magnitude of $u(k)$ in part (b) with the value in Problem 10-6(c), which was $|u(0)| = 0.0441|x(0)|$. Explain the difference.
- (d) Verify all calculations by computer.

10-8. Consider the system of Problem 10.6.

- (a) Find the feedback gain required to minimize the given cost function for the infinite-time problem.
- (b) Find the closed-loop system characteristic equation.
- (c) Find the closed-loop time constant τ as a function of the sample period T .
- (d) Verify all calculations by computer.

10-9. Consider the system of Problem 10.7.

- (a) Find the feedback gain required to minimize the given cost function for the infinite-time problem.
- (b) Find the closed-loop system characteristic equation.
- (c) Find the closed-loop time constant τ as a function of the sample period T .
- (d) Why is the settling time in part (c) less than that found in Problem 10-8(c), where $\tau = 0.410T$.
- (e) Verify all calculations by computer.

10-10. A satellite control system is modeled as shown in Figure P10-10. This system is described in Problem 1-12. For this problem, ignore the sensor gain and let $D(z) = 1$. In addition, $K = 1$, $T = 1$ s, and $J = 4$. As stated in Problem 9-11, a state model for this system is given by

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.125 \\ 0.25 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

where $x_1(k)$ is angular position and $x_2(k)$ is angular velocity.

- (a) Determine the gains required to minimize the cost function

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + 2u^2(k)$$

with $N = 1$. The value of N is chosen to be unity to limit the calculations.

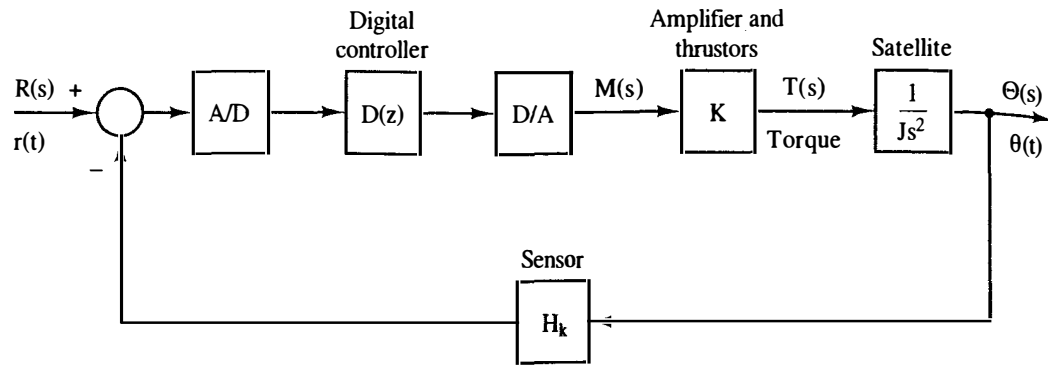


Figure P10-10 Block diagram for a satellite control system.

(b) Find $K(19)$ in part (a) for $N = 20$.

(c) Use a computer to solve part (a) for $N = 20$. Sketch the calculated gains versus k .

10-11. (a) Repeat Problem 10-10 for minimizing the cost function

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + 20u^2(k)$$

Plot the gains on the graph from Problem 10-10(c).

(b) Explain the differences in the gains in part (a) and those in Problem 10-10.

10-12. Consider the satellite control system of Problem 10-10.

(a) By computer, find the feedback gains required to minimize the given cost function for the infinite-time problem.

(b) Find the closed-loop system characteristic equation.

(c) Find the closed-loop time constants as a function of the sample period T .

(d) Verify the calculations in part (b) by computer.

10-13. Consider the satellite control system of Problem 10-11.

(a) By computer, find the feedback gains required to minimize the given cost function for the infinite-time problem.

(b) Find the closed-loop system characteristic equation.

(c) Find the closed-loop time constants as a function of the sample period T .

(d) Compare the time constants in part (c) with the value in Problem 10-12(c), which was $\tau = 3.24T$. Explain the difference.

(e) Verify the calculations in part (b) by computer.

10-14. A chamber temperature control system is modeled as shown in Figure P10-14. This system is described in Problem 1-10. For this problem, ignore the disturbance input, $T = 0.6$ s and let $D(z) = 1$. It was shown in Problem 9-6 that the plant state model is given by

$$x(k+1) = 0.7408x(k) + 1.0368u(k)$$

Let the cost function be given by

$$J_3 = \sum_{k=0}^3 [2x^2(k) + u^2(k)]$$

(a) Calculate the feedback gains required to minimize the cost function, using the partial-differentiation procedure of Section 10.3.

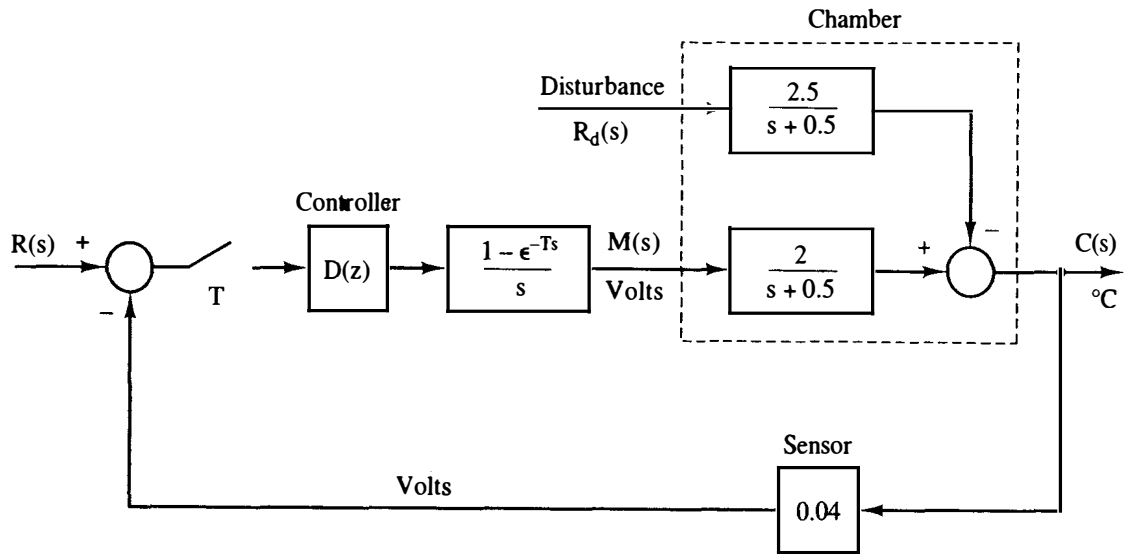


Figure P10-14 Chamber temperature control system.

- (b) Repeat part (a), using the difference-equation approach of Section 10.4.
- (c) Find the maximum magnitude of $u(k)$ as a function of $x(0)$.
- (d) Verify all calculations by computer.

10-15. Given the plant of Problem 10-14, with the cost function

$$J_3 = \sum_{k=0}^3 10x^2(k) + u^2(k)$$

- (a) Calculate the feedback gains required to minimize the cost function, using the difference-equation approach of Section 10.4.
- (b) Find the maximum magnitude of $u(k)$ as a function of $x(0)$.
- (c) Compare the maximum magnitude of $u(k)$ in part (b) with the value in Problem 10-14(c). Explain the difference.
- (d) Verify the results in part (a) by computer.

10-16. Consider the system of Problem 10.14.

- (a) Find the feedback gain required to minimize the given cost function for the infinite-time problem.
- (b) Find the closed-loop system characteristic equation.
- (c) Find the closed-loop time constant τ .
- (d) Verify all calculations by computer.

10-17. Consider the system of Problem 10.15.

- (a) Find the feedback gain required to minimize the given cost function for the infinite-time problem.
- (b) Find the closed-loop system characteristic equation.
- (c) Find the closed-loop time constant τ .
- (d) Compare the gain in part (b) with the value in Problem 10-16(a). Explain the difference.

10-18. Given a general first-order plant described by

$$x(k+1) = Ax(k) + Bu(k)$$

with the cost function

$$J_N = \sum_{k=0}^N [Qx^2(k) + Ru^2(k)]$$

- (a) Show that for $R = 0$, the optimal gain $K(k)$ is constant for all $k \geq 0$.
- (b) Give the input sequence $u(k)$, $k \geq 0$, for part (a), where $u(k)$ is a function of $x(0)$.
- (c) Draw a flow graph of the closed-loop system for part (a).
- (d) Give the closed-loop characteristic equation for part (a).
- (e) How many sample periods are required for $x(k)$, with initial condition $x(0)$, to be driven to zero?

10-19. Suppose that a square-law circuit has the input-output relationship

$$y = kx^2$$

where x is the input voltage and y is the output voltage.

- (a) Derive a least-squares procedure for calculating k , similar to the procedure developed in Section 10.7.
- (b) Experimentation with the circuit yields the following data pairs:

k	x	y
1	0	0.01
2	1.0	1.01
3	2.0	3.98

Find the least-squares value for k for these data.

- (c) To illustrate the effects of erroneous data, suppose that an additional data point, $x = 1.5$ and $y = 3.30$, is taken. From part (b) we know that for $x = 1.5$, y should be approximately 2.25. Hence the additional data point is erroneous, assuming that the data in part (b) are accurate. Find the percent error in \hat{k} using the four data pairs, as compared to the value calculated in part (b).

10-20. (a) A first-order system yields the following input-output measurements:

k	Input	Output
0	10	0
1	10	12.2
2	10	20.1

Find the system transfer function by the least-squares batch procedure, using all the data.

- (b) An additional data pair, for $k = 3$, is measured, with the input equal to zero and the output equal to 31.8. Start the recursive least-squares procedure using the results of part (a), and calculate $\hat{\theta}$ using the additional data pair.
- 10-21.** For a third-order discrete system, suppose that the data pairs $[u(k), y(k)]$, $k = 0, 1, \dots, 5$ are available. In these data, $u(k)$ is the input and $y(k)$ is the output. Write the complete expression for $F^T(N)y(N)$ as in (10-103), using all the data.
- 10-22.** Suppose that a plant is described by

$$x(k+1) = 0.8x(k) + 0.2u(k) + w(k), \quad T = 0.2 \text{ s}$$

$$y(k) = x(k) + v(k)$$

where $w(k)$ and $v(k)$ are random and uncorrelated, with Gaussian distributions, and

$$E[w(k)] = 0, \quad E[w(j)w(k)] = 2\delta_{ij}$$

$$E[v(k)] = 0, \quad E[v(j)v(k)] = \delta_{ij}$$

- (a) Design a Kalman filter for this system. Continue the gain calculations until the gain is approximately constant. Use $M(0) = 2$.
- (b) In part (a), we specified $M(0) = 2$. What are we stating about our estimate of the state $x(0)$?
- (c) Write the difference equations for the steady-state Kalman filter, as designed in part (a).
- (d) Suppose that an LQ design is performed for this plant, with the resulting gain $K = 0.2197$. Find the control-estimator transfer function (see Figure 9-8) for this IH-LQG design.
- (e) Find the closed-loop system characteristic equation for part (d).
- (f) Find the closed-loop system time constants.
- (g) Suppose that the state $x(k)$ is estimated to be 90.1 by the Kalman filter at a certain time kT . Give the three-sigma range about the value 90.1 that will almost certainly contain the true value of $x(k)$ at that time kT .
- (h) If computer facilities are available, find the system phase and gain margins.

10-23. The Kalman filter design in Problem 10-22 resulted in the steady-state filter equations

$$q(k) = \bar{q}(k) + 0.7105[y(k) - \bar{q}(k)]$$

$$\bar{q}(k+1) = 0.8q(k) + 0.2u(k)$$

- (a) Consider the plant and filter to be open-loop; that is, the state estimate is not fed back for control purposes. Suppose that the input $u(k)$ is constant at a value of 10. Find the steady-state values of the plant state $x(k)$ and the plant output $y(k)$, if the random inputs $w(k)$ and $v(k)$ are zero.
- (b) For the conditions specified in part (a), find the steady-state value of $q(k)$, the plant state estimate.
- (c) The steady-state Kalman filter has the property that $Q(z)/U(z) = X(z)/U(z)$. Does this property verify your results in parts (a) and (b)?
- (d) The Kalman filter requires that the average value of $w(k)$ be zero. Suppose that $w(k)$ is constant with a value of 5. Repeat parts (a) and (b), and calculate the percent error in the state estimate.

10-24. Consider the system of Problem 10-22.

- (a) Suppose that all specifications are the same, except that $E[w(j)w(k)] = 0$; that is, there are no random plant disturbances. Design a Kalman filter for this system, and estimate the steady-state Kalman gain from the trends of the calculations. Use $M(0) = 2$.
- (b) Write the difference for the steady-state Kalman filter.
- (c) In part (a), $P(k) \rightarrow 0$ in the steady-state. Hence the errors in the estimation of the state go to zero in the steady-state (perfect estimation). Since the measurements are noisy, how can the estimation error be zero?
- (d) Repeat parts (a), (b), and (c) for the case that $R_w = 2$ and $R_v = 0$. In repeating part (c), the measurements are perfect, but the plant has a random disturbance.

10-25. This problem requires computer calculations. Consider the satellite described in

Problem 10-10. The plant model is given as

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.125 \\ 0.25 \end{bmatrix} u(k) + \begin{bmatrix} 0.01 \\ 0.1 \end{bmatrix} w(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k) + (k)$$

The plant disturbances are caused by random variations in the earth's gravity field. Suppose that $R_w = 1$ and $R_v = 0.01$.

- (a) The measurement $y(k)$ is in the units of angular degrees and is obtained from a stable platform. Describe the accuracy of this measurement; that is, what does the value of R_v tell us about the sensor accuracy?
- (b) Calculate and plot the Kalman gains, as in Figure 10-6, using $\mathbf{M}(0) = \mathbf{I}$.
- (c) The diagonal elements of the steady-state error covariance matrix are

$$\mathbf{P}_{ss} = \begin{bmatrix} 0.00763 & - \\ - & 0.0147 \end{bmatrix}$$

Comment on the steady-state accuracy of this Kalman filter.

- (d) The LQ design of Problem 10.10 yielded the steady-state gains of $\mathbf{K} = [0.5192 \quad 2.103]$. Find the transfer function $D_{ce}(z)$ of the control estimator.
 - (e) Calculate and plot the Nyquist diagram for the closed-loop system opened at the plant input. What are the phase and gain margins?
- 10-26.** This problem requires computer calculations. The system considered here is the classical system of Doyle and Stein [11] to illustrate robustness problems. The system of Doyle and Stein is analog; the discrete model of the system is used here [13]. The sample period, $T = 0.006$ s, was chosen small so that the results approximate those of Doyle and Stein. The plant model is given by

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.999946 & 0.592847E-2 \\ -0.177854E-1 & 0.976233 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.178567E-4 \\ 0.592847E-2 \end{bmatrix} u(k)$$

$$+ \begin{bmatrix} 0.208907 \\ -0.383511 \end{bmatrix} w(k)$$

$$y(k) = [2 \quad 1] \mathbf{x}(k) + v(k)$$

- (a) A pole-placement design is to yield closed-loop poles at $s = -7 \pm j2$. Show that these s -plane poles translate into the desired characteristic equation

$$\alpha_c(z) = z^2 - 1.917602z + 0.919431 = 0$$

- (b) Find the gain matrix required to place the poles at the locations given in part (a).
 - (c) Find the steady-state Kalman filter gains for the case that $R_w = R_v = 166.67$.
 - (d) Find the plant transfer function $G(z)$ and the control-estimator transfer function $D_{ce}(z)$.
 - (e) Using the transfer functions of part (d), plot the Nyquist diagram for the system opened at the plant.
 - (f) Find the system phase and gain margins. Doyle and Stein found the phase margin of the analog system to be approximately 15° .
- 10-27.** Equation (10-88) gives a least-squares estimate for curve fitting. This equation was derived by finding the point at which the slope of the cost function is zero. The

maximum of a function also occurs at the point at which the slope is zero. Show that (10-88) is a minimum, not a maximum, of the cost function.

10-28. Derive (10-106), the weighted least-squares estimate.

10-29. Using (10-106), (10-111), (10-115), and (10-116), derive the equations for the recursive least-squares estimation, (10-118), (10-119), and (10-120).

10-30. For the IH-LQG control system of Figure 10-7, suppose that the plant is single-input single-output, such that \mathbf{B} and \mathbf{C} are vectors. To determine the system robustness, the open-loop transfer functions for the system opened at a and opened at b must be determined.

(a) Find the open-loop transfer function for the system opened at the point a .

(b) Find the open-loop transfer function for the system opened at the point b .

(c) Give the transfer functions for the frequency responses that must be calculated to determine robustness.

Sampled-Data Transformation of Analog Filters

11.1 INTRODUCTION

In Chapters 8, 9, and 10 we presented methods for designing filters, for control systems, in the digital domain. Many times an application may require the transformation of an existing analog design to the digital domain. This requirement may result when an existing continuous control system is being replaced or updated with a digital version. Digital circuits eliminate many reliability problems and are less susceptible to electronic noise and electromagnetic radiation. In other cases the system designers may have more experience in designing continuous controllers and wish to design their filters in the s -domain and then transform them into the z -domain.

Consequently, in this chapter we first present some basic ideas about sampled-data transformations. Next we review the fundamentals of designing Butterworth, Bessel, transitional, Chebyshev, and elliptic analog filters. These filter design methods are useful for implementing low-pass, high-pass, band-pass, and band-stop filters to be employed in control systems which have special requirements for processing sensor signals, eliminating noise frequency bands, and the like. Finally, we apply the sampled-data transforms to a typical analog filter.

11.2 SAMPLED-DATA TRANSFORMATIONS

Sampled-data transformations are the techniques one uses to obtain numerical solutions to integral and differential equations. Any linear system's transfer function

may be written as

$$G(s) = \frac{Y(s)}{X(s)}$$

$Y(s)$ = Laplace transform of the output

$X(s)$ = Laplace transform of the input

Alternatively, the relationship between input and output may be described as a differential or integral equation. Numerical methods may be employed to solve these equations; these methods approximate the integral and differential equations by difference equations. As we have seen previously, the difference equations may be represented by a discrete transfer function. The complete process is illustrated in Figure 11-1.

Numerical Approximations

Several numerical approximation techniques will now be presented, with some for differentiation and some for integration.

Backward difference. The backward difference is a simple technique that replaces the derivative of a function by

$$\frac{d}{dt} y(t) \doteq \frac{y(t) - y(t - T)}{T}$$

(See Figure 11-2.)

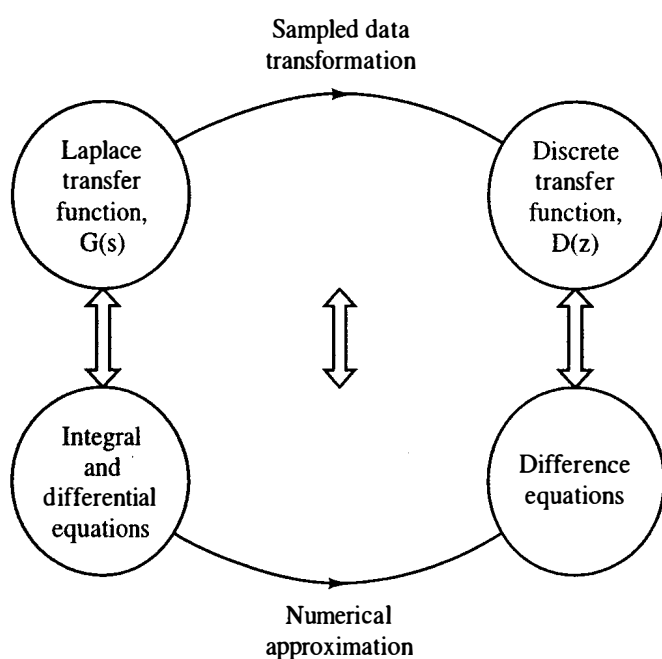


Figure 11-1 Relation between numerical approximations and sampled data transformations.

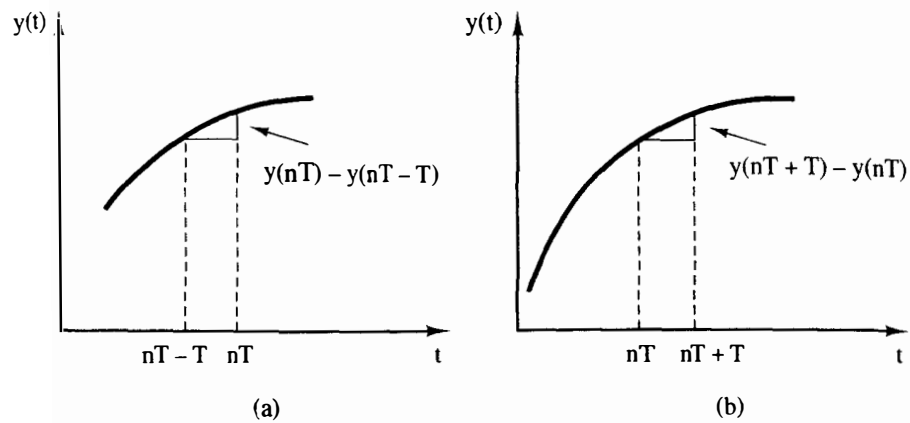


Figure 11-2 Difference approximations: (a) backward difference; (b) forward difference.

In the Laplace domain

$$sY(s) \doteq \frac{Y(s) - e^{-sT} Y(s)}{T} + y(0+)$$

If $y(0+)$ is small, then

$$s \doteq \frac{1 - e^{-sT}}{T}$$

$$s \doteq \frac{1 - z^{-1}}{T}$$

Hence

$$D(z) = G(s)|_{s = (1 - z^{-1})/T} \quad (11-1)$$

Example 11.1

Find a discrete approximation for

$$G(s) = \frac{s}{s + a}$$

$$Y(s) = G(s)X(s)$$

$$sY(s) + aY(s) = sX(s)$$

or

$$\frac{d}{dt} y(t) + ay(t) = \frac{d}{dt} x(t)$$

Now let

$$\frac{d}{dt} y(t) = \frac{y(t) - y(t - T)}{T}$$

$$\frac{d}{dt} x(t) = \frac{x(t) - x(t - T)}{T}$$

Therefore,

$$\frac{y(t) - y(t - T)}{T} + ay(t) = \frac{x(t) - x(t - T)}{T}$$

Evaluating at $t = nT$ yields

$$y(nT) = \frac{1}{1 + Ta} [x(nT) - x(nT - T) + y(nT - T)]$$

Employing the techniques of Chapter 2, we have

$$D(z) = \frac{1}{1 + Ta} \frac{1 - z^{-1}}{1 - \frac{1}{1 + Ta} z^{-1}}$$

An alternative solution employs (11-1) as follows:

$$\begin{aligned} D(z) &= \left. \frac{s}{s + a} \right|_{s = (1 - z^{-1})/T} \\ &= \frac{\frac{1 - z^{-1}}{T}}{a + \frac{1 - z^{-1}}{T}} \\ &= \frac{1 - z^{-1}}{aT + 1 - z^{-1}} \\ &= \frac{1}{1 + aT} \frac{1 - z^{-1}}{1 - \frac{1}{1 + aT} z^{-1}} \end{aligned}$$

Forward difference. A similar numerical technique approximates

$$\frac{d}{dt}y(t) \doteq \frac{y(t + T) - y(t)}{T}$$

(See Figure 11-2.)

This represents the equivalent Laplace domain approximation

$$sY(s) \doteq \frac{e^{sT} Y(s) - Y(s)}{T} + y(0+)$$

and if $y(0+)$ is neglected,

$$\begin{aligned} s &\doteq \frac{e^{sT} - 1}{T} \\ &\doteq \frac{z - 1}{T} \end{aligned}$$

Hence

$$D(z) = G(s)|_{s = (z - 1)/T} \quad (11-2)$$

Example 11.2

Find a discrete version of $G(s)$ using the forward difference.

$$\begin{aligned}
 G(s) &= \frac{s}{s+a} \\
 D(z) &= \left. \frac{s}{s+a} \right|_{s=(z-1)/T} \\
 &= \frac{(z-1)/T}{[(z-1)/T] + a} \\
 &= \frac{1-z^{-1}}{1+(aT-1)z^{-1}}
 \end{aligned}$$

Rectangular rule. Suppose now that we try some numerical approximations to integrals and compare results. The idea here is to represent $G(s)$ as

$$G(s) = \frac{\alpha_0 + \alpha_1 s^{-1} + \alpha_2 s^{-2} + \cdots + \alpha_n s^{-n}}{1 + \beta_1 s^{-1} + \beta_2 s^{-2} + \cdots + \beta_n s^{-n}} \quad (11-3)$$

Each s^{-1} represents an integrator in the s -domain. Hence, if we can replace each integrator by its digital equivalent

$$s^{-1} = f(z)$$

or

$$s = \frac{1}{f(z)} = g(z)$$

a digital equivalent of $G(s)$ will be produced.

Left-side rule. Let us determine the numerical approximation for

$$y(t) = \int_0^t x(t) dt$$

Assume that the upper limit of the integral is $t = nT$. Hence

$$y(nT) = \int_0^{nT} x(t) dt \quad (11-4)$$

Figure 11-3a illustrates the rectangular rule using the left side of the rectangles. Hence

$$\begin{aligned}
 y(nT) &= T \sum_{i=0}^{n-1} x(iT) \\
 y(nT + T) &= T \sum_{i=0}^n x(iT) = T \sum_{i=0}^{n-1} x(iT) + Tx(nT) \\
 &= y(nT) + Tx(nT)
 \end{aligned}$$

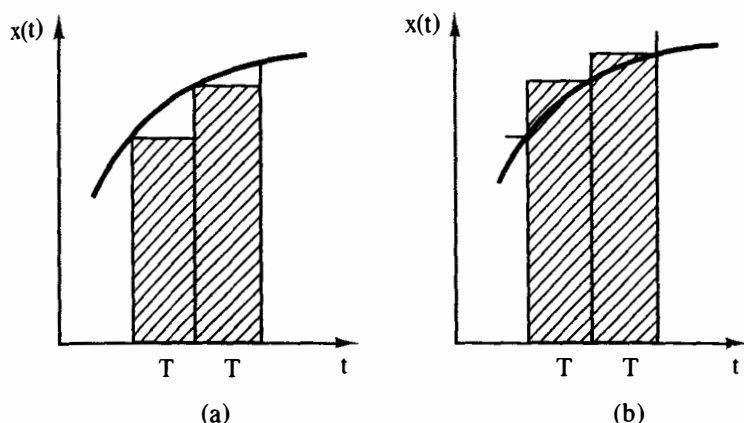


Figure 11-3 Rectangular rule:
(a) left-side rule; (b) right-side rule.

Therefore, using the results of Chapter 2, the transfer function $Y(z)/X(z)$ is

$$D(z) = \frac{Tz^{-1}}{1 - z^{-1}}$$

$$= \frac{T}{z - 1}$$

Hence we have approximated the integration transfer function

$$\frac{1}{s} \doteq \frac{T}{z - 1} = f(z)$$

which gives the same results as (11-2) for the forward difference.

Right-side rule. Figure 11-3b illustrates the use of the right side of the rectangle in approximating (11-4). Therefore,

$$y(nT) = T \sum_{i=1}^n x(iT)$$

$$y(nT + T) = T \sum_{i=1}^{n+1} x(iT) = T \sum_{i=1}^n x(iT) + Tx(nT + T)$$

$$= y(nT) + Tx(nT + T)$$

Letting $n = n - 1$ yields

$$y(nT) = y(nT - T) + Tx(nT)$$

Hence the transfer function is

$$D(z) = \frac{T}{1 - z^{-1}}$$

Consequently, we have approximated the integrator

$$\frac{1}{s} \doteq \frac{T}{1 - z^{-1}} = f(z)$$

which yields the identical result of (11-1) for the backward difference.

Trapezoidal rule. The trapezoidal rule takes the average of the left and right sides of the rectangles in Figure 11-3. Hence

$$\begin{aligned} y(nT) &= \frac{T}{2} \sum_{i=0}^{n-1} [x(iT) + x(iT + T)] \\ &= \frac{1}{2} \left[T \sum_{i=0}^{n-1} x(iT) + T \sum_{i=1}^n x(iT) \right] \end{aligned}$$

Using the results of the rectangular rule, we see that the transfer function $Y(z)/X(z)$ is

$$\begin{aligned} D(z) &= \frac{1}{2} \left[\frac{Tz^{-1}}{1 - z^{-1}} + \frac{T}{1 - z^{-1}} \right] \\ &= \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} \end{aligned}$$

Thus we have approximated

$$\frac{1}{s} \doteq \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} = f(z)$$

This approximation is the familiar bilinear z -transform.

Simpson's rule. Simpson's rule evaluates (11-4) by the formula

$$y(nT) = \frac{T}{3} [x(0) + 4x(T) + 2x(2T) + \cdots + 4x(nT - T) + x(nT)]$$

But

$$y(nT + 2T) = y(nT) + \frac{T}{3} [x(nT) + 4x(nT + T) + x(nT + 2T)]$$

Letting $n = n - 2$, the transfer function is

$$D(z) = \frac{T}{3} \frac{1 + 4z^{-1} + z^{-2}}{1 - z^{-2}}$$

Hence we have approximated

$$\frac{1}{s} \doteq \frac{T}{3} \frac{1 + 4z^{-1} + z^{-2}}{1 - z^{-2}} = f(z)$$

Impulse invariance [1]. Suppose that we want to find a discrete equivalent filter for the Laplace transfer function $G(s)$. Further suppose that we desire the impulse response of the discrete equivalent to match that of the analog filter as shown in Figure 11-4; that is, we desire impulse invariance:

$$g(nT) = d(nT)$$

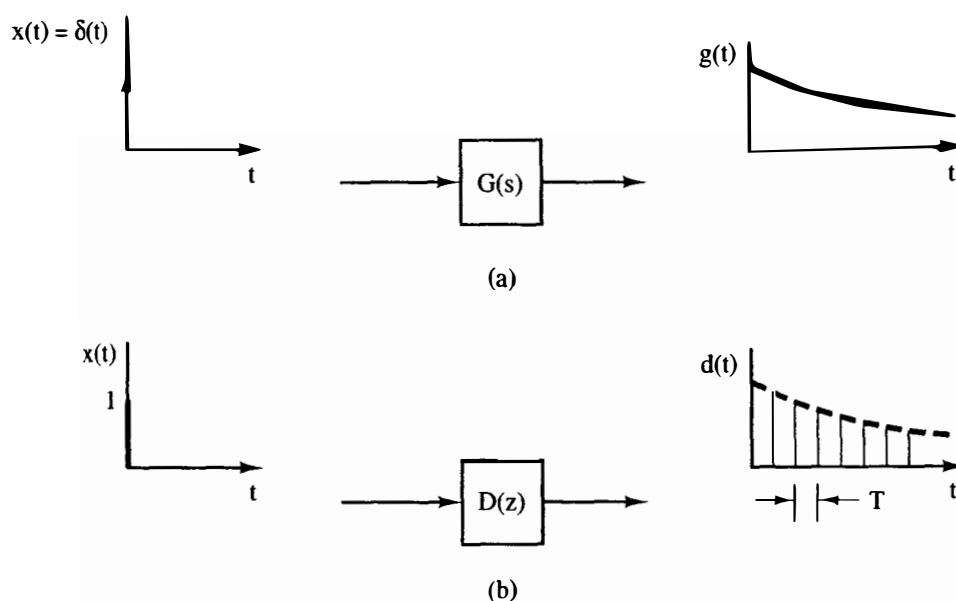


Figure 11-4 Impulse invariance: (a) analog filter; (b) digital filter.

Then

$$\begin{aligned} D(z) &= \sum_{i=0}^{\infty} d(nT)z^{-n} \\ &= \sum_{i=0}^{\infty} g(nT)z^{-n} \\ &= G(z) \end{aligned}$$

which is the standard z -transform. Hence for impulse invariance,

$$D(z) = \mathcal{Z}[G(s)] = G(z)$$

the digital approximation is just the standard z -transform of $G(s)$.

Impulse invariant integrator. Let us find the digital equivalent of an analog integrator using impulse invariance and the models of Figure 11-5. We know that

$$G(z) = \mathcal{Z}\left[\frac{1}{s}\right] = \frac{1}{1 - z^{-1}}$$

and that

$$G_{h0}(s) = \frac{1 - e^{-Ts}}{s} \doteq T$$

for small values of T . Hence

$$\frac{Y_d(z)}{X(z)} = \frac{T}{1 - z^{-1}}$$

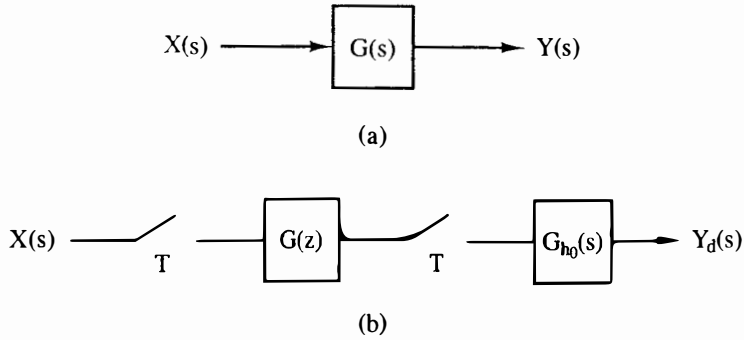


Figure 11-5 Impulse invariant integrator: (a) analog integrator; (b) digital integrator.

and we have again approximated

$$\frac{1}{s} \doteq \frac{T}{1 - z^{-1}} = f(z)$$

Therefore, the backward difference, the right-side rectangular rule, and the impulse invariant integrator all indicate (11-1) as their equivalent sampled-data transformation.

Step invariance [2]. In Figure 11-4 a digital filter $D(z)$ that preserves the impulse response of an analog filter $G(s)$ was derived, and the result was the standard z -transform of $G(s)$. Suppose that instead of preserving the impulse response we preserve the step response. That is, the step response of $G(s)$ is set equal to the step response of $D(z)$ on a sample-by-sample basis. Then

$$\left(\frac{1}{1 - z^{-1}}\right)D(z) = \mathcal{Z}\left[\left(\frac{1}{s}\right)G(s)\right]$$

or

$$D(z) = (1 - z^{-1})\mathcal{Z}\left[\frac{G(s)}{s}\right]$$

Tables for $D(z)$ may be found in Ref. 2.

Example 11.3

Consider the step invariant integrator

$$G(s) = \frac{1}{s}$$

and

$$\begin{aligned} D(z) &= (1 - z^{-1})\mathcal{Z}\left[\frac{1}{s^2}\right] \\ &= (1 - z^{-1})\left[\frac{Tz^{-1}}{(1 - z^{-1})^2}\right] \\ &= \frac{Tz^{-1}}{1 - z^{-1}} \end{aligned}$$

which yields the same approximation derived earlier for the forward difference.

Mapping Functions Summary

As a result of our analysis of some elementary numerical approximation techniques we have identified several sampled-data mapping functions.

Standard z -transform. The standard z -transform yields an impulse-invariant filter. The mapping function for this transformation is

$$s = \frac{1}{T} \ln z \quad (11-5)$$

This mapping was defined in Chapter 4.

Backward difference [1]. The backward-difference approximation for the solution of differential equations provides the following mapping:

$$s = \frac{1 - z^{-1}}{T} \quad (11-6)$$

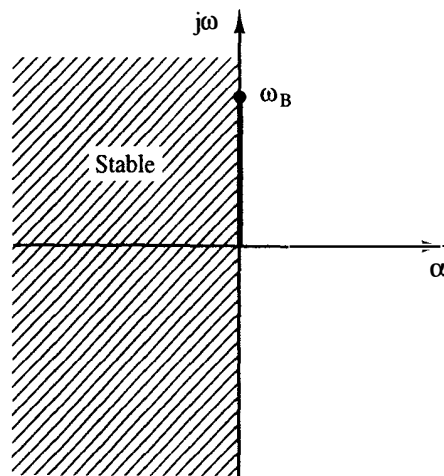
Solving for z yields

$$z = \frac{1}{1 - Ts}$$

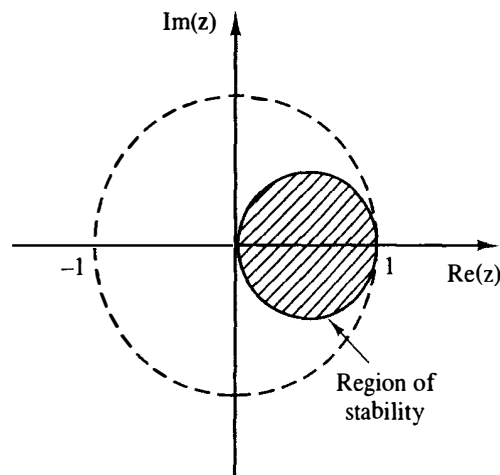
Now substituting the frequency contour $s = j\omega$ produces

$$\begin{aligned} z &= \frac{1}{1 - j\omega T} \\ &= \frac{\frac{1}{2}(1 - j\omega T) + \frac{1}{2}(1 + j\omega T)}{1 - j\omega T} \\ &= \frac{1}{2} \left[1 + \frac{1 + j\omega T}{1 - j\omega T} \right] \\ &= \frac{1}{2} [1 + e^{j2 \tan^{-1}(\omega T)}] \\ &= \frac{1}{2} + \frac{1}{2} e^{j\theta} \end{aligned}$$

Consequently, the left half of the s -plane maps inside the unit circle of the z -plane as shown in Figure 11-6. Hence stable analog filters will always result in stable digital equivalents. In fact, some unstable analog filters give stable digital ones. A major disadvantage of this mapping is seen in the frequency-response contour. The $j\omega$ -axis in the s -plane does not map to the unit circle in the z -plane. Hence, as we get farther from $s = 0$ (or $z = 1$), the more degraded will be our desired frequency response. Thus we must decrease T (increase f_s) to improve the approximation.



(a)



(b)

Figure 11-6 Mapping $z = 1/(1 - sT)$:
(a) s -plane; (b) z -plane.

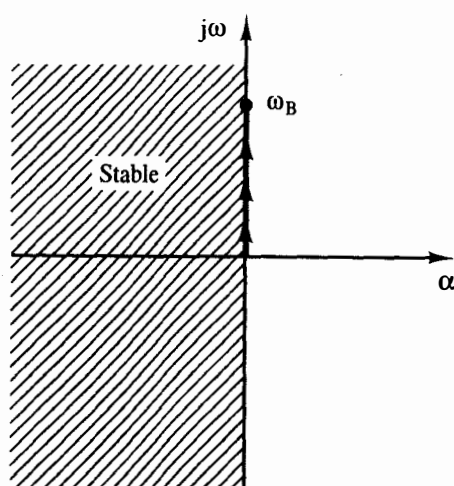
Forward difference. The forward-difference approximation suggested the mapping

$$s = \frac{z - 1}{T}$$

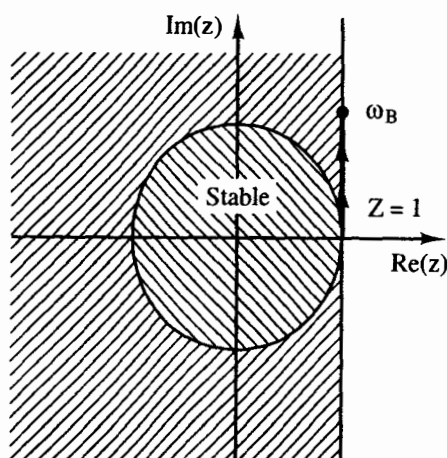
Solving for z yields

$$z = 1 + Ts \quad (11-7)$$

This mapping function is shown in Figure 11-7. Note that the left half-plane in the s -domain maps to the region to the left of $z = 1$ in the z -plane. But the interior of the unit circle represents the stability region in the z -plane. Consequently, some stable analog filters will give *unstable* digital ones. Unstable analog filters will also be unstable digital ones under this mapping. Yet a further disadvantage is that the frequency contour in the z -plane does not follow the unit circle. Hence this is an undesirable mapping.



(a)



(b)

Figure 11-7 Mapping $z = 1 + Ts$: (a) s -plane; (b) z -plane.

Bilinear z -transformation. The trapezoidal integration approximation led to the sampled-data mapping

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

Solving for z yields

$$z = \frac{(2/T) + s}{(2/T) - s} \quad (11-8)$$

This mapping is illustrated in Figure 11-8. In Chapter 2 we employed this transform for a different purpose. Note here that the entire left half s -plane maps to the interior of the unit circle in the z -plane. Hence all stable analog filters will result in stable digital ones. Also, the $j\omega$ -axis in the s -plane maps to the unit circle in the z -plane. However, the entire $j\omega$ -axis maps *onto* the unit circle, which causes a mismatching

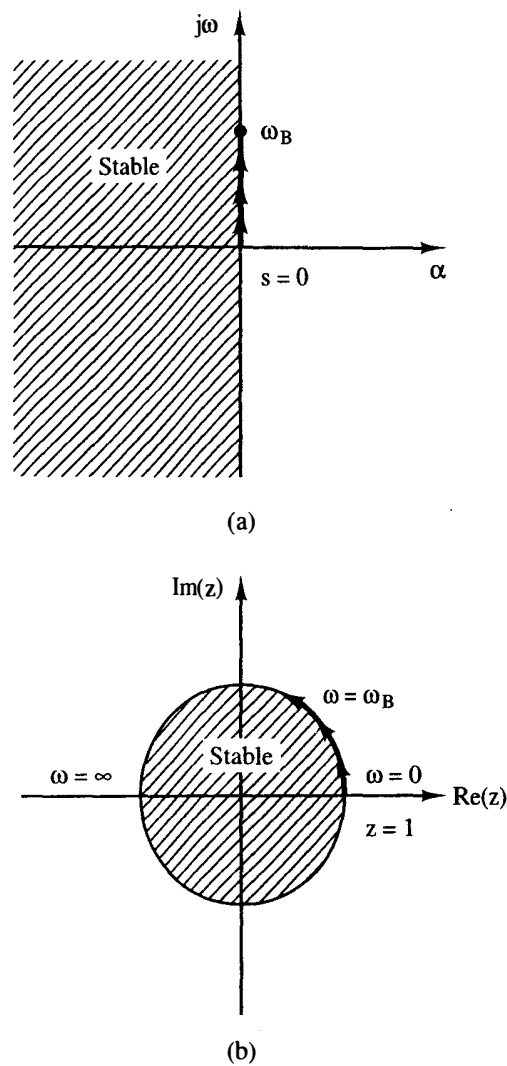


Figure 11-8 Bilinear z -transform; (a) s -plane; (b) z -plane.

of frequencies. This is a direct result of the characteristic that for a digital filter

$$z = 1 \rightarrow \omega = 0, \quad z = j1 \rightarrow \omega = \frac{\omega_s}{4}, \quad z = -1 \rightarrow \omega = \frac{\omega_s}{2}$$

as required by (11-5). For the bilinear z -transform the frequencies in the z -plane (ω_D) are related to frequencies in the s -plane (ω_A) by

$$\frac{j\omega_A T}{2} = \frac{e^{j\omega_D T} - 1}{e^{j\omega_D T} + 1} = \frac{2j \sin(\omega_D T/2)}{2 \cos(\omega_D T/2)}$$

or

$$\omega_D = \frac{2}{T} \tan^{-1} \frac{\omega_A T}{2} \quad (11-9)$$

(See Figure 11-9.) Correction for this frequency-scale warping may be accomplished

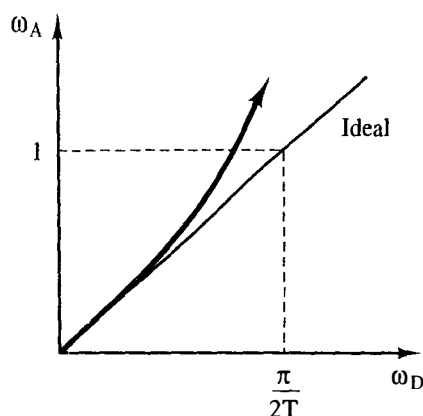


Figure 11-9 Change in frequency scale for the bilinear z -transform.

by redesigning (prewarping) the critical frequencies of the desired transfer function $G(s)$ before applying the bilinear z -transform.

This transformation maps circles and straight lines in the s -plane to circles in the z -plane. It works well for frequency characteristics that are piecewise linear. It also ensures that *no* frequency aliasing can occur in the transfer-function frequency characteristic because the $+j\omega$ -axis does map onto the upper half of the unit circle. Hence the bilinear z -transform is quite popular.

Matched z -transforms [3]. The standard z -transform of $G(s)$ of (11-3) requires a partial-fraction expansion of $G(s)$ in order to complete the mapping

$$\frac{1}{s + u} \rightarrow \frac{1}{1 - e^{-uT} z^{-1}}$$

For the purpose of simplifying the calculations, the matched z -transform maps the poles and zeros of $G(s)$ to the z -plane as follows:

$$s + \alpha \rightarrow 1 - z^{-1} e^{-\alpha T} \quad (11-10)$$

Hence the matched z -transform of (11-3) is

$$\begin{aligned} G(z) &= G(s) \Big|_{\substack{s + a_i = 1 - z^{-1} e^{-a_i T} \\ s + b_j = 1 - z^{-1} e^{-b_j T}}} \\ &= K \frac{\prod_{i=1}^m (1 - z^{-1} e^{-a_i T})}{\prod_{j=1}^n (1 - z^{-1} e^{-b_j T})} \end{aligned} \quad (11-11)$$

where K is adjusted to give the desired gain at dc ($z = 1$). This transform *matches* the poles and zeros in the s - and z -planes. Note that the poles of this transform are identical with those of the standard z -transform but that the zeros are different. Because of this difference, the matched z -transform may be used on nonbandlimited inputs. If $G(s)$ has no zeros, it is sometimes necessary to multiply $(1 + z^{-1})^N$, N an integer, times (11-11).

Other Transforms

In general, any transformation that maps the stable region of the s -plane into the stable region of the z -plane may be used. It is helpful for the $j\omega$ -axis in the s -plane to map to the z -plane's unit circle. Another important property is that rational functions $G(s)$ should be transformed into rational functions $D(z)$ so that the proper difference equations may be determined for realization.

Simpson's rule. The Simpson's rule approximation suggested that the mapping

$$s = \frac{3}{T} \frac{1 - z^{-2}}{1 + 4z^{-1} + z^{-2}}$$

be used as a transformation. Note that a second-order function $G(s)$ will transform to a fourth-order $D(z)$. This is undesirable from an implementation viewpoint.

(w, v)-transform [4]. In some applications the system transfer function $G(s, z, z^\alpha)$ may be a function of s , $z = e^{Ts}$, and z^α , where $0 < \alpha < 1$. If all initial conditions are zero and

$$w = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

$$v(\alpha) = 1 - \alpha(1 - z^{-1}) + \frac{\alpha(\alpha - 1)}{2}(1 - z^{-1})^2$$

then for a system described by

$$Y(s) = G(s, z, z^{-\alpha})X(s)$$

its z -transform will be

$$Y(z) = G(w, z, v(\alpha)) \left[X(z) - \frac{x(0)}{1 + z^{-1}} \right]$$

If $x(0) = 0$, then

$$D(z) = G(s, z, z^{-\alpha}) \Big|_{\substack{s=w \\ z^{-\alpha}=v(\alpha)}}$$

Example 11.4

Scott [5] has shown that a desirable phase-locked loop has the transfer function

$$G(s) = \frac{10}{s + 10z^{-0.5}}$$

Using the (w, v) transform to find a digital equivalent, if $x(0) = 0$,

$$D(z) = \frac{10}{s + 10z^{-0.5}} \Big|_{\substack{s=w = (2/T)[(1 - z^{-1})/(1 + z^{-1})] \\ z^{-0.5}=v(0.5)}}$$

$$\begin{aligned} v(0.5) &= 1 - 0.5(1 - z^{-1}) + \frac{0.5(-0.5)}{2}(1 - z^{-1})^2 \\ &= 0.375 + 0.75z^{-1} - 0.125z^{-2} \end{aligned}$$

$$D(z) = \frac{5T(1 + z^{-1})}{(1 + 1.875T) - (1 - 5.625T)z^{-1} + (3.125T)z^{-2} - (0.625T)z^{-3}}$$

z-Forms. Boxer and Thaler [6-8] have developed a transformation that preserves, in the digital filter, the time response of its analog ancestor. The method, called z-forms, expresses

$$s = \frac{\ln z}{T}$$

as a series by substituting

$$\ln z = 2 \left[u + \frac{u^3}{3} + \frac{u^5}{5} + \dots \right]$$

where

$$u = \frac{z - 1}{z + 1}$$

Therefore,

$$\begin{aligned} s^{-1} &= \frac{T}{\ln z} \\ &= \frac{T}{2} \frac{u^{-1} - u/3 - 4u^3}{45 - 44u^5/945 - \dots} \end{aligned} \quad (11-12)$$

To preserve the order of the filter (make the digital filter the same order as the analog one), the positive powers of u in (11-12) are truncated as follows:

$$\begin{aligned} s^{-1} &\approx \frac{T}{2} [u^{-1}] \\ &= \frac{T}{2} \frac{z + 1}{z - 1} \end{aligned}$$

For higher negative powers of s , s^{-k} , the series in (11-12) is raised to the power k , and the positive powers of u are then truncated. For example, for $k = 2$,

$$\begin{aligned} s^{-2} &\approx \frac{T^2}{4} \left[u^{-2} - \frac{2}{3} \right] \\ &= \frac{T^2}{12} \left[\frac{z^2 + 10z + 1}{(z - 1)^2} \right] \end{aligned}$$

For $k = 3$,

$$\begin{aligned} s^{-3} &\approx \frac{T^3}{8} [u^{-3} - u^{-1}] \\ &= \frac{T^3}{2} \left[\frac{z^2 + z}{(z - 1)^3} \right] \end{aligned}$$

The results of these calculations are listed in Table 11-1 [8]. Table 11-1 is used in transforming an analog filter to a digital one as follows:

1. Express $G(s)$ as a rational fraction in powers of s^{-1} by dividing the numerator and denominator by s^n , where n is the degree of the denominator polynomial.
2. Substitute for each s^{-k} its z -form from Table 11-1.
3. Divide the resulting expression by T to obtain $D(z)$.

Example 11.5

Let

$$G(s) = \frac{1}{s^3 + s^2 + s}$$

Dividing the numerator and denominator by s^3 yields

$$G(s) = \frac{s^{-3}}{1 + s^{-1} + s^{-2}}$$

Substituting from Table 11-1 the proper z -forms and dividing by T produces

$$D(z) = \frac{6T^2(z^2 + z)}{(12 + 6T + T^2)z^3 - (36 + 6T - 9T^2)z^2 + (36 - 6T - 9T^2)z - (12 - 6T + T^2)}$$

If $T = 1$ s.

$$D(z) = \frac{6z^2 + 6z}{19z^3 - 33z^2 + 21z - 7}$$

TABLE 11-1 z -FORMS

Power of s	z -Transform $\mathcal{Z}[s^{-k}] = N_k(z)/D_k(z)$	z -Forms
s^{-1}	$z/(z - 1)$	$(T/2)(z + 1)/(z - 1)$
s^{-2}	$Tz/(z - 1)^2$	$(T^2/12)(z^2 + 10z + 1)/(z - 1)^2$
s^{-3}	$(T^2/2)(z^2 + z)/(z - 1)^3$	$(T^3/2)(z^2 + z)/(z - 1)^3$
s^{-4}	$(T^3/6)(z^3 + 4z^2 + z)/(z - 1)^4$	$(T^4/6)(z^3 + 4z^2 + z)/(z - 1)^4 - T^4/720$
s^{-5}	$(T^4/24)(z^4 + 11z^3 + 11z^2 + z)/(z - 1)^5$	$(T^5/24)(z^4 + 11z^3 + 11z^2 + z)/(z - 1)^5$
s^{-k-1}	$(Tz/k)[kN_k(z) - (z - 1)N'_k(z)]/(z - 1)^{k+1}$	$T\mathcal{Z}[s^{-k-1}] + T^{k+1}A_{k+1}(-1)^{(k+1)/2+1}$

Notes:

1. A_k is the Bernoulli coefficient [9].
2. $N'_k(z)$ represents $dN_k(z)/dz$.

11.3 REVIEW OF CONTINUOUS FILTER DESIGN [10]

The design of continuous filters can be accomplished by first designing several low-pass filter transfer functions $G(s)$, called prototype or normalized designs; the prototypes have a critical or break frequency of 1 rad/s. The prototype is used to realize a filter for a given specification by using the following frequency transformations:

$$\begin{aligned} \text{Low pass: } s &\rightarrow \frac{\hat{s}}{\omega_u} \\ \text{Band pass: } \hat{s} &\rightarrow \frac{s^2 + \omega_u \omega_l}{s(\omega_u - \omega_l)} \\ \text{Band stop: } s &\rightarrow \frac{s(\omega_u - \omega_l)}{s^2 + \omega_u \omega_l} \\ \text{High pass: } s &\rightarrow \frac{\omega_u}{s} \end{aligned} \tag{11-13}$$

where ω_u is the upper cutoff and ω_l is the low cutoff.

Five prototype filters are discussed in this section: Butterworth, Bessel, transitional, Chebyshev, and elliptical designs.

Butterworth

The Butterworth approximation to the ideal low-pass filter is defined by the squared frequency magnitude function

$$|G(\omega)|^2 = \frac{1}{1 + (\omega^2)^n} \tag{11-14}$$

where n is the order of the filter. The Laplace-transfer function is given by

$$G(s)G(-s) = \frac{1}{1 + (-1)^n s^{2n}}$$

or

$$G(s) = \prod_{j=1}^n \frac{1}{s + b_j}$$

where

$$b_j = \exp\left[-i\pi\left(\frac{1}{2} + \frac{2j-1}{2n}\right)\right], \quad i = \sqrt{-1}$$

Example 11.6

Design a low-pass filter to be used in a digital control system for antialiasing. That is, the filter should be placed in front of the analog-to-digital converter (ADC) in the control loop to ensure that the frequency content of the signal reaching the ADC is



greatly attenuated above half the sampling frequency. Let the sampling frequency be 1000 Hz.

We may use MATLAB to design this filter [11]. First try a second-order Butterworth prototype:

```
>> [z1,p1,k1]=buttap(2);[num1,den1]=zp2tf(z1,p1,k1)
```

```
num1 =  
0 0 1
```

```
den1 =  
1.0000e+00 1.4142e+00 1.0000e+00
```

Next use the low-pass frequency transformation of (11-13) to move the cutoff frequency to 400 Hz:

```
>> [num2,den2]=lp2lp(num1,den1,2*pi*400)
```

```
num2 =  
0 0 6.3165e+06
```

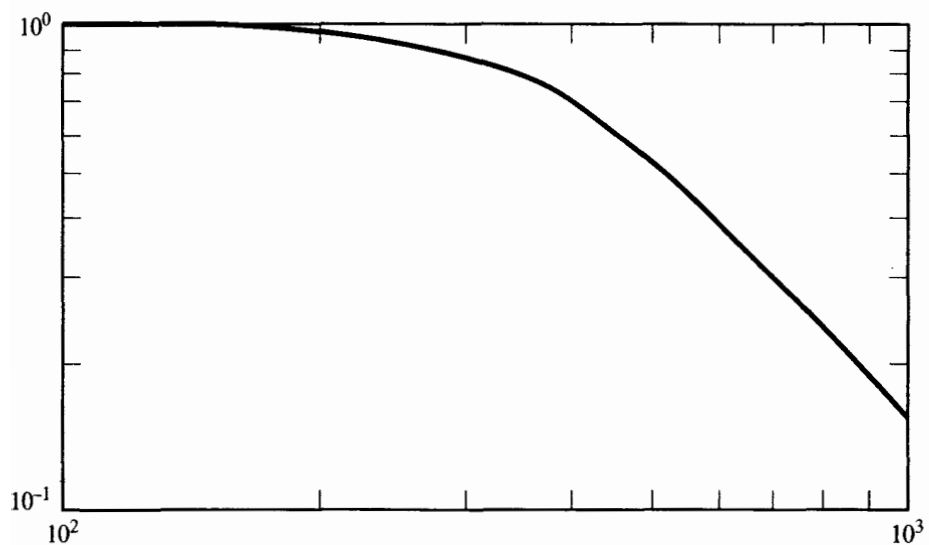
```
den2 =  
1.0000e+00 3.5543e+03 6.3165e+06
```

Then compute the frequency response of the filter and check the dB value at half the sampling rate (500 Hz):

```
>> f=logspace(2,3); w=2*pi*f; h=freqs(num2,den2,w); mag=abs(h); loglog(f,mag);  
h500=freqs(num2,den2,2*pi*500); mag500=abs(h500); db500 = 20*log10(mag500)
```

```
db500 =
```

```
-5.3674e+00
```

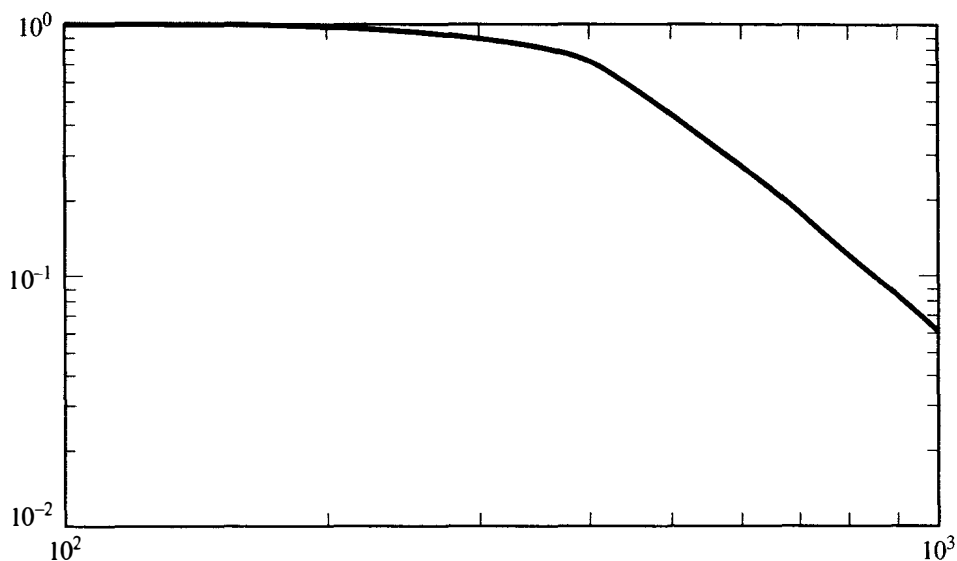


Some frequency aliasing will occur using this low-pass filter. We can reduce aliasing by increasing the order of the Butterworth prototype. For the third-order case:

```
>> [z1,p1,k1] = buttap(3); [num1,den1] = zp2tf(z1,p1,k1); [num2,den2] =
lp2lp(num1,den1,2*pi*400); f = logspace(2,3); w = 2*pi*f; h =
freqs(num2,den2,w); mag = abs(h); loglog(f,mag); h500 =
freqs(num2,den2,2*pi*500); mag500 = abs(h500); db500 = 20*log10(mag500)

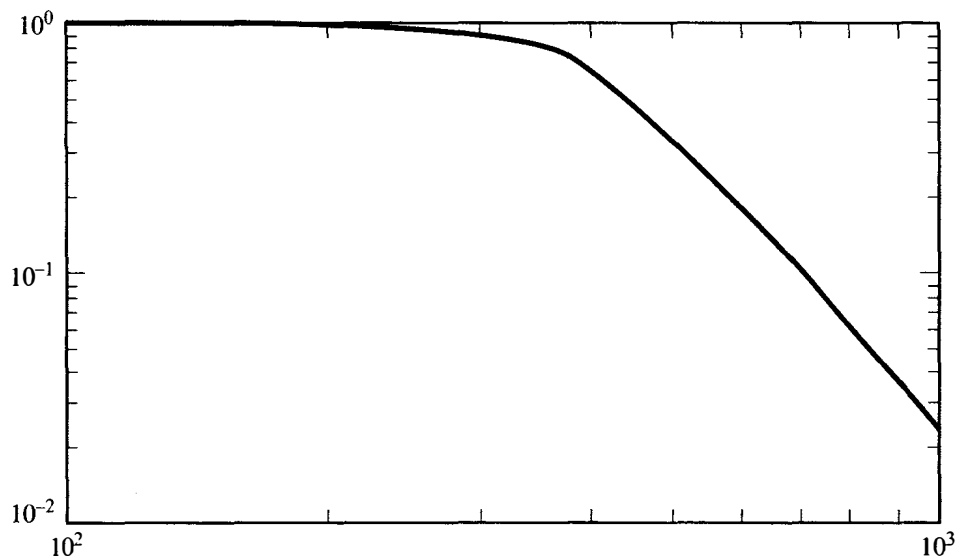
db500 =

-6.8257e+00
```



For the fourth-order case:

```
>> [z1,p1,k1] = buttap(4); [num1,den1] = zp2tf(z1,p1,k1); [num2,den2] =
lp2lp(num1,den1,2*pi*400); f = logspace(2,3); w = 2*pi*f; h =
```



```

freqs(num2,den2,w); mag = abs(h); loglog(f,mag); h500 =
freqs(num2,den2,2*pi*500); mag500 = abs(h500); db500 = 20*log10(mag500)

db500 =

-8.4264e+00

```

For our design, let us choose the fourth-order design to achieve better performance in antialiasing. Therefore, we may printout the coefficients of our design:

```

>> num2, den2

num2 =

    0  2.7285e-12  3.7253e-09  -2.2888e-05  3.9899e+13

den2 =

 1.0000e+00  6.5675e+03  2.1566e+07  4.1484e+10  3.9899e+13

```

So our final design is

$$G(s) = \frac{2.7285 \times 10^{-12} s^3 + 3.7253 \times 10^{-9} s^2 - 2.2888 \times 10^{-5} s + 3.9899 \times 10^{13}}{s^4 + 6.5675 \times 10^3 s^3 + 2.1566 \times 10^7 s^2 + 4.1484 \times 10^{10} s + 3.9899 \times 10^{13}}$$

Bessel

The Bessel filter approximation for the linear delay function $\epsilon^{-\tau s}$ may be written

$$G(s) = \frac{K_0}{B_n(s)} \quad (11-15)$$

where K_0 is a constant term and $B_n(s)$ are Bessel polynomials.

$$B_0 = 1$$

$$B_1 = s + 1$$

$$\vdots$$

$$B_n = (2n - 1)B_{n-1} + s^2 B_{n-2}$$

The roots of $B_n(s)$ are normalized using the factor $(K_0)^{1/n}$.

Transitional

The transitional filter combines roots of the n th-order Butterworth and normalized Bessel filters according to a transitional factor TF. Let

$$r_j = \text{magnitude of } j\text{th transitional pole}$$

$$r_{1j} = \text{magnitude of } j\text{th Bessel pole}$$

θ_j = angle of j th transitional pole

θ_{1j} = angle of j th Bessel pole

θ_{2j} = angle of j th Butterworth pole

the poles of the transitional filter are then described by

$$\begin{aligned} r_j &= r_{1j} \text{TF} \\ \theta_j &= \theta_{2j} + \text{TF}(\theta_{1j} - \theta_{2j}) \end{aligned} \quad (11-16)$$

Chebyshev

Chebyshev filters exhibit better cutoff characteristics for lower-order filters than do the designs above. Chebyshev type I and type II filters are defined by

$$|G_1(\omega)|^2 = \frac{1}{1 + \epsilon^2 T_n^2(\omega)} \quad (11-17)$$

and

$$|G_2(\omega)|^2 = \frac{1}{1 + \epsilon^2 \left[\frac{T_n(\omega_r)}{T_n(\omega_r/\omega)} \right]^2} \quad (11-18)$$

where $T_n(\omega) = \cos(n \cos^{-1} \omega)$, $0 \leq \omega \leq 1$

$= \cosh(n \cosh^{-1} \omega)$, $\omega > 1$

$$T_0(\omega) = 1$$

$$T_1(\omega) = \omega$$

$$T_2(\omega) = 2\omega^2 - 1$$

$$T_3(\omega) = 4\omega^3 - 3\omega$$

The order of the filter n is determined by specifying inband ripple E and the lowest frequency at which a loss of a decibels is achieved. Hence

$$\begin{aligned} \epsilon &= (10^{E/10} - 1)^{1/2} \\ A^2 &= 10^{a/10} \end{aligned} \quad (11-19)$$

and

$$n = \frac{\cosh^{-1} \sqrt{A^2 - 1/\epsilon}}{\cosh^{-1}(\omega_r)}$$

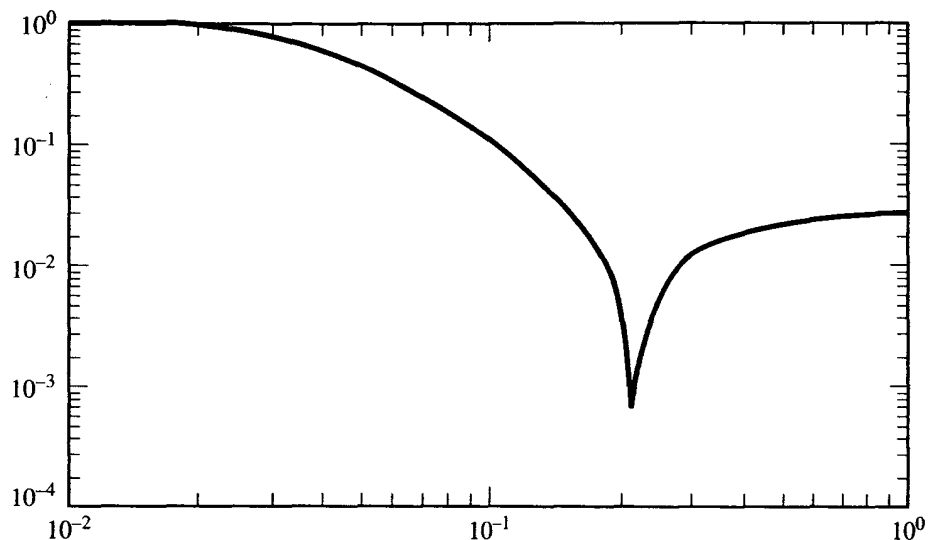
In (11-19), the variables E , a , or ω_r must be adjusted so that the n will be an integer. The type I filter differs from the type II in that the type I exhibits equiripple in the pass band while type II has equiripple in the stop band.

Example 11.7

Design a high-pass filter to be used in a digital control system. The filter is to eliminate dc from the controller input signal because an integrator is included ($1/(1 - z^{-1})$ factor in the denominator) in the design. Dc offsets in the controller input signal can cause the controller output to saturate, opening the control loop and causing instability. The final design should have at least 30 dB band rejection below 0.1 Hz, with less than 5 dB attenuation at 0.2 Hz. The sampling frequency is 1000 Hz.

We may use MATLAB to design this filter. Since the specification calls for a band rejection of greater than 30 dB, try a second-order Chebyshev type II prototype with 30-dB stop-band attenuation (plot the frequency response):

```
>> [z1,p1,k1] = cheb2ap(2,30); [num1,den1] = zp2tf(z1,p1,k1); f = logspace(-2,0); w = 2*pi*f; h = freqs(num1,den1,w); mag = abs(h); loglog(f,mag)
```



Next use the high-pass frequency transformation of (11-13) to move the cutoff frequency to 0.1 Hz. The MATLAB command is

```
>> [num2,den2] = 1p2hp(num1,den1,2*pi*.1)
```

```
num2 =  
1.0000e+00      0 1.9739e-01
```

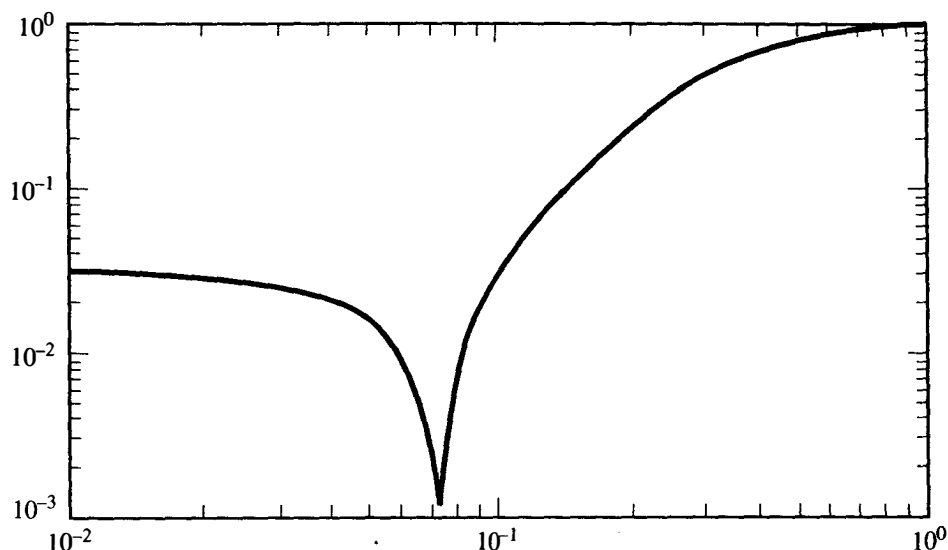
```
den2 =  
1.0000e+00 3.4770e+00 6.2421e+00
```

Then compute the frequency response of the filter and check the dB value at 0.1 and 0.2 Hz:

```
>> f = logspace(-2,0); w = 2*pi*f; h = freqs(num2,den2,w); mag = abs(h);  
loglog(f,mag); h1 = freqs(num2,den2,2*pi*.1); mag1 = abs(h1); db1 =  
20*log10(mag1), h2 = freqs(num2,den2,2*pi*.2); mag2 = abs(h2); db2 =  
20*log10(mag2)
```

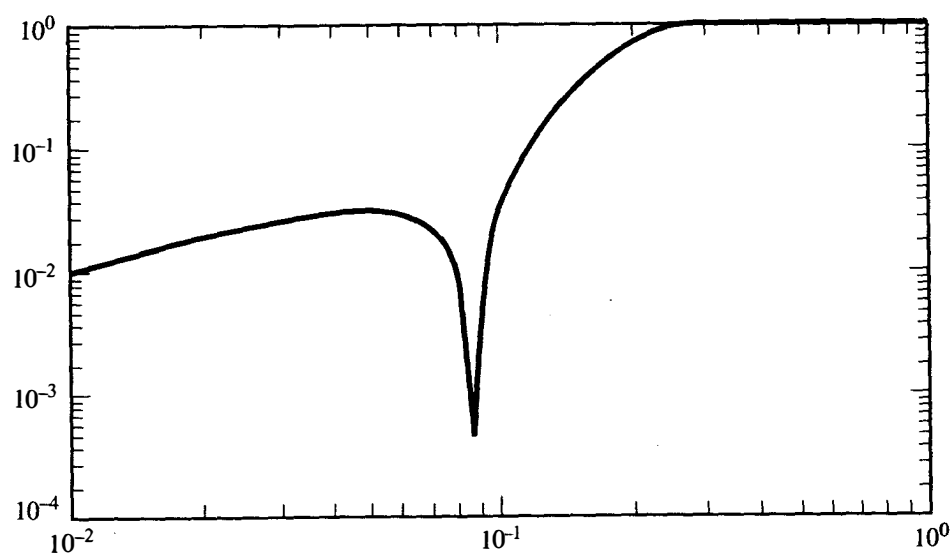
db1 =
-30

db2 =
-1.3302e+01



The response is down 30 dB at 0.1 Hz, as expected, but is down over 13 dB at 0.2 Hz. So next try a third-order prototype design:

```
>> [z1,p1,k1] = cheb2ap(3,30); [num1,den1] = zp2tf(z1,p1,k1); [num2,den2] =  
lp2hp(num1,den1,2*pi*.1); f = logspace(-2,0); w = 2*pi*f; h =  
freqs(num2,den2,w); mag = abs(h); loglog(f,mag); h1 =  
freqs(num2,den2,2*pi*.1); mag1 = abs(h1); db1 = 20*log10(mag1); h2 =  
freqs(num2,den2,2*pi*.2); mag2 = abs(h2); db2 = 20*log10(mag2)
```



```

db1 =
-3.0000e+01
db2 =
-3.9407e+00

```

This meets the specifications of the required filter. Thus the final design is

```

>> num2, den2
num2 =
1.0000e+00      0  2.9609e-01      0
den2 =
1.0000e+00  2.3454e+00  3.0466e+00  1.9600e+00

```

and the transfer function may be expressed as

$$G(s) = \frac{s^3 + 0.29609s}{s^3 + 2.3454s^2 + 3.0466s + 1.9600}$$

Elliptic

The elliptic filter has equiripple in both the pass and stop bands. Hence this type of design usually achieves the desired frequency response with a lower-order n than any of the types described above. The elliptic filter is determined by

$$|G(\omega)|^2 = \frac{1}{1 + \epsilon^2 \psi_n^2(\omega)} \quad (11-20)$$

where

$$\psi_n = \begin{cases} \operatorname{sn} \left[n \frac{K(k_1)}{K(k)} \operatorname{sn}^{-1}(\omega; k); k_1 \right], & n \text{ odd} \\ \operatorname{sn} \left[K(k_1) + N \frac{K(k_1)}{K(k)} \operatorname{sn}^{-1}(\omega; k); k_1 \right], & n \text{ even} \end{cases}$$

with

$$\chi = \int_0^\omega \frac{d\omega}{[(1 - \omega^2)(1 - k^2 \omega^2)]^{1/2}} = \text{elliptic integral of the first kind}$$

$\operatorname{sn}[\chi; k] = \omega = \text{Jacobian elliptic function}$

$K(k) = \text{complete elliptic integral of the first kind}$

$$= \int_0^{\pi/2} \frac{d\phi}{(1 - k^2 \sin^2 \phi)^{1/2}}$$

$$k = \frac{1}{\omega_r}$$

$$k_1 = \epsilon(A^2 - 1)^{-1/2}$$

$$\epsilon = (10^{E/10} - 1)^{1/2}$$

$$A^2 = 10^{a/10}$$

where e , a , and ω_r were defined for the Chebyshev filter; the order n is found by

$$n = \frac{K(k_1')K(k)}{K(k_1)K(k')}$$

with $k' = (1 - k^2)^{1/2}$ and $k_1' = (1 - k_1^2)^{1/2}$. The result of any of the five design methods results in a Laplace transfer function $G(s)$ for the desired frequency response.

11.4 TRANSFORMING ANALOG FILTERS

Earlier in this chapter we described numerous sampled-data transformations that may be employed to achieve the goal of producing a digital approximation for an analog filter. However, we recommend that the standard z -transform, the bilinear z -transform, the matched z -transform, or z -forms be used in most practical applications. In this section we demonstrate the standard, bilinear, and matched z -transforms. The reader is referred to Refs. 6 and 8 for a discussion of the accuracy of the z -forms method.

Standard z -Transform

The problem of converting a continuous filter to a discrete one is presented in Figure 11-10. This figure is an expanded version of the ideas presented in Figure 11-5. Figure 11-10a illustrates the basic analog filter. Figure 11-10b shows the impulse invariance implications of the standard z -transform. If

$$Y_a^*(s) = Y_b^*(s)$$

then

$$Y_a^*(s) = D^*(s)X^*(s)$$

But

$$Y_a^*(s) = (G(s)X(s))^*$$

Impulse invariance implies that

$$D(z) \doteq G(z)$$

or

$$D^*(s) \doteq G^*(s)$$

Consequently,

$$(G(s)X(s))^* \doteq G^*(s)X^*(s)$$

This is the first assumption.

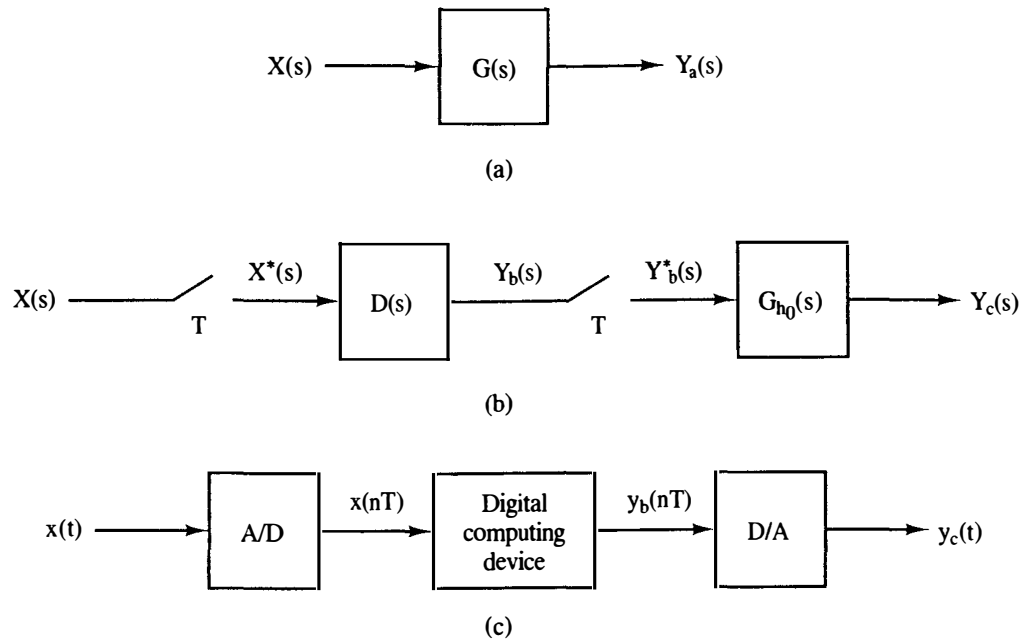


Figure 11-10 Transforming a continuous filter by the standard z -transform: (a) analog design; (b) impulse invariant model of an analog filter; (c) computational model.

Next, consider Figure 11-10c regarding the digital implementation of the analog filter. First, it is desirable that

$$y_c(nT) = y_a(nT)$$

But

$$Y_c(s) = G_{h0}(s)Y_b^*(s)$$

where

$$G_{h0}(s) = \frac{1 - e^{-Ts}}{s}$$

$$Y_c(s) = D^*(s)X^*(s)G_{h0}(s)$$

Hence, since $G_{h0}^*(s) = 1$,

$$\begin{aligned} Y_c^*(s) &= [D^*(s)X^*(s)G_{h0}(s)]^* \\ &= D^*(s)X^*(s)G_{h0}^*(s) = D^*(s)X^*(s) \end{aligned}$$

Consequently,

$$D(z) = G(z)$$

is the transfer function of the digital computer and we may write

$$D(z) = \sum_{k=1}^n \frac{R_k}{1 - z^{-1}e^{-Tb_k}} \quad (11-21)$$

An important point to remember is that if a zero-order-hold device is not being used, we adjust the gain of our filter by a factor T . Remember that:

1. Sampling increases the gain by a factor $1/T$ [see (3-35)].
2. The zero-order hold restores the gain by a factor T since

$$G_{ho}(s) = \frac{1 - e^{-Ts}}{s} \approx \frac{1 - (1 - Ts)}{s} = T$$

for small T .

Consequently, for comparing the frequency responses, one should compute $G(j\omega)$ and $TD(e^{j\omega T})$. Note that the standard z -transform can be used only on bandlimited signals ($f < f_s/2$).

Bilinear z -Transform

The bilinear z -transform may be used to obtain a discrete equivalent of $G(s)$ as follows:

$$D(z) = G'(s) \Big|_{s = (2/T)(1 - z^{-1})/(1 + z^{-1})} \quad (11-22)$$

where $G'(s)$ is a continuous filter whose critical frequencies differ from $G(s)$ by

$$f'_s = \frac{1}{\pi T \tan(\pi f_c T)} \quad (11-23)$$

Relation (11-23) is used *before* the continuous filter $G(s)$ is designed. The new filter $G'(s)$ is designed instead and then transformed to the z -plane by (11-22). The bilinear z -transform is a bandlimiting transformation with relatively flat magnitude characteristics in the pass and stop bands. However, the time response will be considerably different.

Matched z -Transform

The matched z -transform matches the poles and zeros of the discrete function to those of the continuous one. The digital equivalent of the $G(s)$ function is calculated as follows:

$$D(z) = G(s) \Big|_{\substack{s + a_i = 1 - z^{-1} e^{-a_i T} \\ s + b_j = 1 - z^{-1} e^{-b_j T}}} \quad (11-24)$$

If $G(s)$ has no zeros, it is sometimes necessary to multiply (11-24) by $(1 + z^{-1})^N$, N is an integer.

Summary

The standard z -transform is suitable only for bandlimited functions, while the bilinear and matched z -transforms are suitable for all filter types. The matched

z -transform requires $G(s)$ in factored form; standard, in partial-fraction form; and bilinear, in prewarped frequency form. The standard z -transform preserves the shape of the impulse-time response; the matched, the shape of the frequency response; and the bilinear, the flat magnitude gain-frequency response characteristics. An example filter is designed and discretized in the following example.

Example 11.8



Design a 50-Hz notch filter to be used in a digital control system in a research laboratory in France. Power-line interference is to be eliminated digitally by inserting this 50-Hz notch filter in cascade with the digital controller. Because 50 Hz is in the range of control frequencies for the system, the bandwidth of rejected frequencies should be kept quite narrow. The 50-Hz rejection should be at least 100 dB while keeping the attenuation at 49 and 51 Hz to 3 dB or less. The attenuation at below 48 and above 52 Hz should be less than 1 percent.

The design procedure to be followed is first to design a Butterworth prototype, transform it to a band-stop filter with a center frequency of 50 Hz and bandwidth of 4 Hz, and then use the bilinear z -transform with a matching frequency of 50 Hz to achieve the final digital filter design. Using MATLAB we can design a Butterworth prototype and translate it to a band-stop filter. Start with a second-order Butterworth prototype:

```
>> [z1,p1,k1] = buttap(2); [num1,den1] = zp2tf(z1,p1,k1);
```

Next transform it to a band-stop filter with a center frequency of 50 Hz and a bandwidth of 2 Hz:

```
>> [num2,den2] = lp2bs(num1,den1,2*pi*50, 2*pi*2);
```

Then find the frequency response of the bandstop filter to determine if it meets the original specifications:

```
f = [10 48 49 50 51 52 100]; w = 2*pi*f; h = freqs(num2,den2,w); mag = abs(h)
```

```
mag =
```

```
Columns 1 through 6
```

```
1.0000e+00 9.7241e-01 7.1425e-01 0 7.0011e-01 9.6785e-01
```

```
Column 7
```

```
1.0000e+00
```

Note that the response is down about 3 dB (0.707) at 49 and 51 Hz, but at 48 and 52 Hz the attenuation is about 3 percent when less than 1 percent is desired. So repeat the design procedure using a third-order Butterworth prototype:

```
>> [z1,p1,k1] = buttap(3); [num1,den1] = zp2tf(z1,p1,k1); [num2,den2] =
1p2bs(num1,den1,2*pi*50,2*pi*2); f = [10 48 49 50 51 52 100]; w = 2*pi*f; h =
freqs(num2,den2,w); mag = abs(h)
```

```
mag =
```

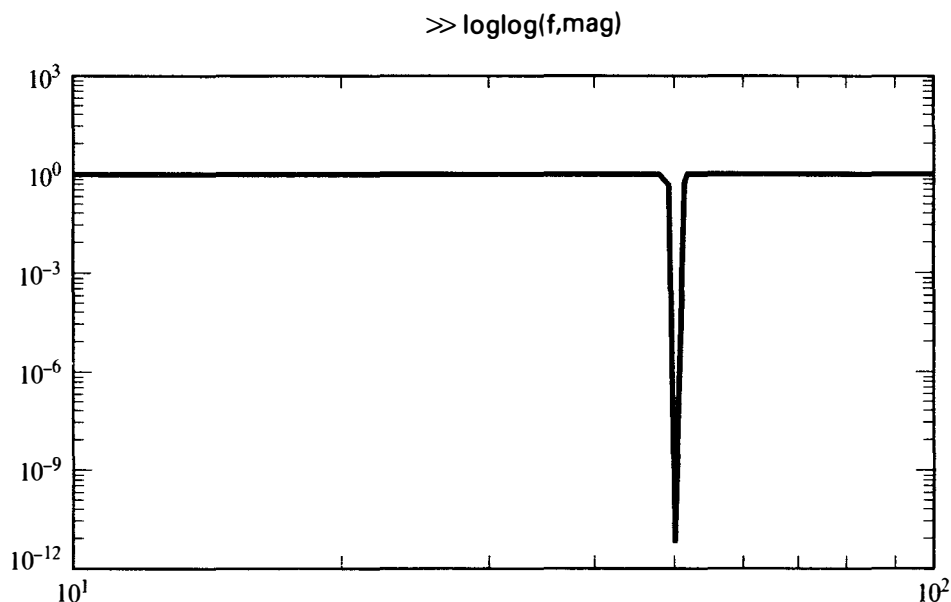
```
Columns 1 through 6
```

```
1.0000e+00 9.9317e-01 7.1779e-01 8.1263e-12 6.9658e-01 9.9134e-01
```

```
Column 7
```

```
1.0000e+00
```

This design meets the specifications, so plot the frequency to verify the tabular results:



Now the analog filter must be transformed to the z-domain. Use the bilinear transformation and a sampling frequency of 1000 Hz:

```
>> fs = 1000; [numd,dend] = bilinear(num2,den2,fs); hd = freqz(numd,dend,w/fs);
magd = abs(hd), loglog(f,magd)
```

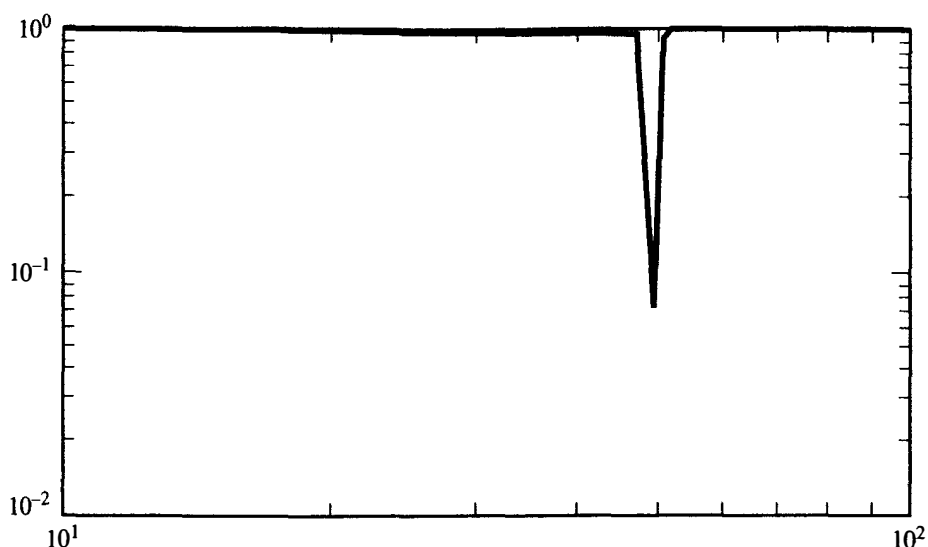
```
magd =
```

```
Columns 1 through 6
```

```
1.0000e+00 9.7696e-01 2.2447e-01 7.0587e-02 9.4424e-01 9.9745e-01
```

```
Column 7
```

```
1.0000e+00
```

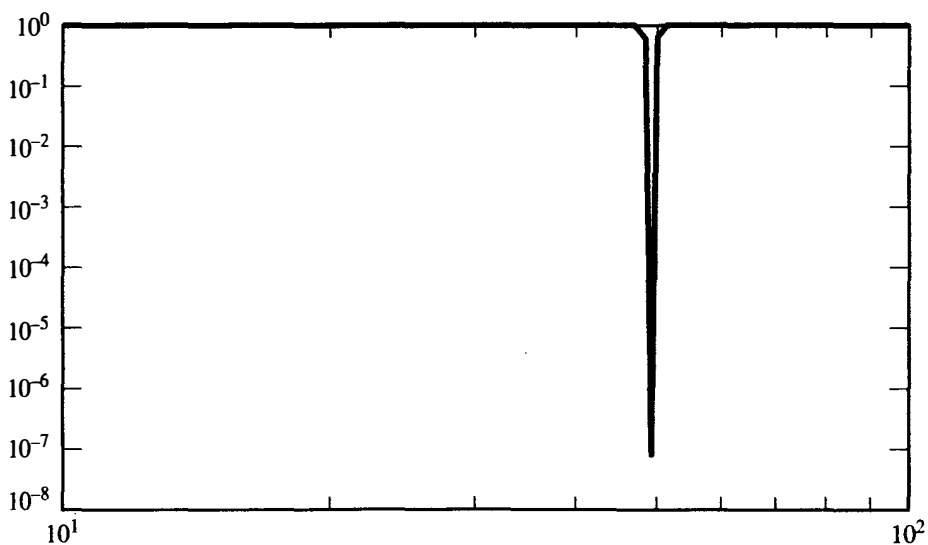
Note that the bilinear transformation has shifted the notch in the filter, so that the resulting design does not meet the original specifications. We may force the bilinear transformation to match digital and analog frequencies at one specified point by adding this matching frequency to the argument list as shown below. Choose 50 Hz as the matching frequency:

```
>> fs = 1000; [numd,dend] = bilinear(num2,den2,fs,50); hd =
freqz(numd,dend,w/fs); magd = abs(hd), loglog(f,magd)
```

```
magd =
```

```
Columns 1 through 6
```

```
1.0000e+00 9.9378e-01 7.3445e-01 5.4440e-08 7.1448e-01 9.9217e-01
```



Column 7

1.0000e+00

This design meets the original specifications. In summary:

>> num2, den2

num2 =

Columns 1 through 6

1.0000e+00 -2.2782e-13 2.9609e+05 -4.4970e-08 2.9223e+10 -2.2192e-03

Column 7

9.6139e+14

den2 =

Columns 1 through 6

1.0000e+00 2.5133e+01 2.9640e+05 4.9630e+06 2.9254e+10 2.4482e+11

Column 7

9.6139e+14

>> numd, dend

numd =

Columns 1 through 6

9.8781e-01 -5.6414e+00 1.3703e+01 -1.8098e+01 1.3703e+01 -5.6414e+00

Column 7

9.8781e-01

dend =

Columns 1 through 6

1.0000e+00 -5.6877e+00 1.3759e+01 -1.8097e+01 1.3647e+01 -5.5954e+00

Column 7

9.7577e-01

Therefore, the analog and digital filter transfer functions are

$$G(s) = \frac{s^6 - 2.2782 \times 10^{-13}s^5 + 2.9609 \times 10^5s^4 - 4.4970 \times 10^{-8}s^3}{s^6 + 2.5133 \times 10^1s^5 + 2.9640 \times 10^5s^4 + 4.4630 \times 10^6s^3} \\ \frac{+ 2.9223 \times 10^{10}s^2 - 2.2192 \times 10^{-3}s + 9.6139 \times 10^{14}}{+ 2.9254 \times 10^{10}s^2 + 2.4482 \times 10^{11}s + 9.6139 \times 10^{14}}$$

$$D(z) = 0.98781 \frac{1.0 - 5.7110z^{-1} + 13.872z^{-2} - 18.321z^{-3} + 13.872z^{-4} - 5.7110z^{-5} + z^{-6}}{1.0 - 5.6877z^{-1} + 13.759z^{-2} - 18.097z^{-3} + 13.647z^{-4} - 5.59540z^{-5} + 0.97577z^{-6}}$$

11.5 SUMMARY

In this chapter we have summarized most of the sampled-data transforms that are commonly used for transforming analog filters to digital filters. We have presented derivations of some of the transforms in order to give the reader a perspective on their relative accuracy and importance. Design examples using MATLAB were presented to illustrate some of the most important principles.

REFERENCES

1. A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.
2. C. P. Newman and C. S. Baradello, "Digital Transfer Functions for Microcomputer Control," *IEEE Trans. Syst. Man Cybern.*, Vol. SMC-9, No. 12, pp. 856-860, Dec. 1979.
3. R. M. Golden, "Designing Digital Filters, z-Transforms, and Fourier Analysis," *Proc. Natl. Electron. Conf.*, St. Charles, IL, June 1969.
4. C. A. Halijak, "The (w, v) -Transform," *Proc. IEEE Region 3 Conv.*, Knoxville, TN, Apr. 10-12, 1972, pp. C4.1-3.
5. R. E. Scott, "An Improved Phase Lock Loop Derived from Ideal Single Sideband Modulation," Ph.D. dissertation, University of Denver, Denver, CO, June 1966, pp. 20-27.
6. R. Boxer and S. Thaler, "A Simplified Method of Solving Linear and Nonlinear Systems," *Proc. IRE*, Vol. 44, pp. 89-101, Jan. 1956.
7. R. Boxer, "Frequency Analysis of Computer Systems," *Proc. IRE*, Vol. 43, pp. 228-229, Feb. 1955.
8. R. Boxer, "A Note on Numerical Transform Calculus," *Proc. IRE*, Vol. 45, pp. 1401-1406, Oct. 1957.
9. C. H. Richardson, *An Introduction to the Calculus of Finite Differences*. New York: D. Van Nostrand Co., 1954, pp. 45-46.
10. A. S. Sedra and P. O. Brackett, *Filter Theory and Design: Active and Passive*. Beaverton, OR: Matrix Publishers, Inc., 1978.
11. *The Student Edition of MATLAB*. Englewood Cliffs, NJ: Prentice Hall, 1992.

PROBLEMS

11-1. Given $G(s) = (s + 2)/(s^2 + 4s + 3)$ and $T = 1$ s, find the (a) standard z-transform; (b) bilinear z-transform; (c) matched z-transform.

11-2. Given $G(s) = [(s + 1)(s + 20)]/[(s + 2)(s + 10)]$, find an equivalent $D(z)$ using:

$$(a) s = \frac{1 - z^{-1}}{T}$$

$$(b) s = \frac{z - 1}{T}$$

$$(c) s = \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

$$(d) s = e^{Ts}$$

$$(e) s + \alpha \Rightarrow 1 - z^{-1} e^{-\alpha T}$$

11-3. Design a Chebyshev I prototype filter for $n = 3$, $a = 6$ dB, and $E = 1$ dB.

11-4. Using the prototype of Problem 11-3, design a low-pass analog filter with

$$\omega_u = 2\pi(100)$$

Use the bilinear z-transform to find a digital equivalent ($f_s = 1000$ Hz). Set the dc gain to 1.

11-5. Given $G(s) = G_1(s)G_2(s)$, where

$$G_1(s) = \frac{98,596s^2}{s^4 + 154.186s^3 + 491,994s^2 + 30,348,629s + 3.8884 \times 10^{10}}$$

$$G_2(s) = \frac{314s}{s^2 + 155.15s + 197,192}$$

(a) Plot the frequency response for $G(s)$.

(b) Find an equivalent $D(z)$ using the bilinear z-transform. Plot its frequency response.

(c) Repeat part (b) using the matched z-transform.

11-6. Repeat Problem 11-5 using (a) the backward difference; (b) the forward difference.

11-7. Compare the results of Problems 11-5 and 11-6.

11-8. Given a Chebyshev I prototype frequency function

$$|G_1(\omega)|^2 = \frac{1}{\omega^4 - \omega^2 + 1.25}$$

show that

$$G_1(s) = \frac{1}{s^2 + 1.112s + 1.118}$$

11-9. An analog PID controller may be written as

$$G(s) = K_P + K_I \frac{1}{s} + K_D s$$

Use the backward-difference mapping and find a digital equivalent.

11-10. Repeat Problem 11-9 using the trapezoidal mapping function.

11-11. Compare the answers to Problems 11-9 and 11-10 to the PID controller of (8-52). Which mapping functions are used in (8-52)?

- 11-12. Use MATLAB to improve the design of the filter in Example 11.1. Achieve an attenuation greater than 30 dB of frequencies above 500 Hz.
- 11-13. Can Example 11.2 be designed using a Chebyshev type I prototype? If so, give the filter's transfer function.
- 11-14. Repeat Example 11.3 for 60-Hz power-line noise rejection.

Digital Filter Structures

12.1 INTRODUCTION

Up to this point we have been concerned with finding a transfer function $D(z)$ in the z -domain that is to perform digital filtering and control operations. The transfer function may be represented in general by

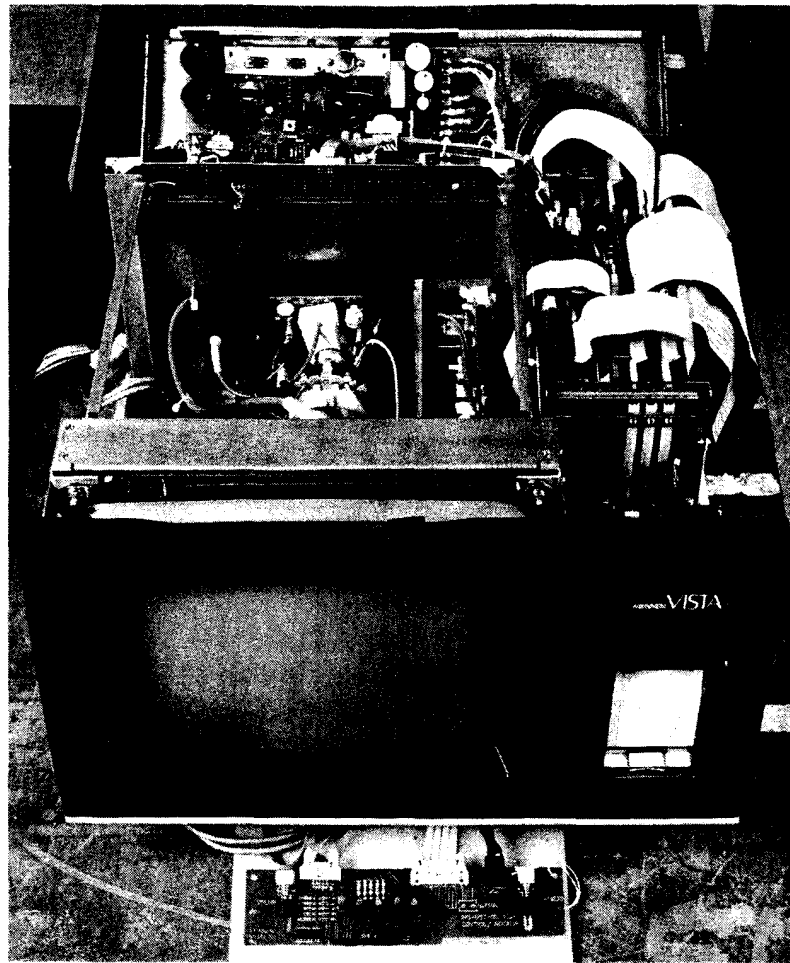
$$D(z) = \frac{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}}{1 + b_1 z^{-1} + \cdots + b_n z^{-n}} \quad (12-1)$$

where a_i and b_i are real coefficients and n is the maximum of the orders of the denominator and numerator polynomials. Either polynomial may have zero coefficients in the higher-order terms so that (12-1) may describe the general case.

The purpose of this chapter is to describe block diagram realizations of (12-1) using time-delay elements (represented by z^{-1}), adders, and multipliers. Each different block diagram is called a *filter structure*. Needless to say, there exist countless structures for (12-1) and we will describe only a few of the more important ones. In particular, we describe direct-form structures, second-order modules, cascaded modules, paralleled modules, and ladder structures, and suggest others.

12.2 DIRECT STRUCTURES

Direct structures for digital filters are those in which the real coefficients, a_i and b_i of (12-1), appear as multipliers in the block diagram implementation.



Nurse station for monitoring multipatient ECG, arrhythmia, and hemodynamic data, based on a single-board computer. (Courtesy of Digital Equipment Corporation.)

First Direct Structure

Suppose that we represent

$$D(z) = \frac{\sum_{i=0}^n a_i z^{-i}}{\sum_{i=0}^n b_i z^{-i}} \quad (12-2)$$

where $b_0 = 1$. If $X(z)$ is the filter input and $Y(z)$ is the output, then

$$\frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^n a_i z^{-i}}{\sum_{i=0}^n b_i z^{-i}}$$

If an intermediate variable, say $M(z)$, is introduced,

$$\frac{Y(z)}{M(z)} \frac{M(z)}{X(z)} = \frac{\sum_{i=0}^n a_i z^{-i}}{\sum_{i=0}^n b_i z^{-i}}$$

such that

$$\frac{Y(z)}{M(z)} = \sum_{i=0}^n a_i z^{-i}$$

$$\frac{X(z)}{M(z)} = \sum_{i=0}^n b_i z^{-i}$$

Hence

$$X(z) = \sum_{i=0}^n b_i z^{-i} M(z)$$

or

$$M(z) = X(z) - \sum_{i=1}^n b_i z^{-i} M(z)$$

and

$$Y(z) = \sum_{i=0}^n a_i z^{-i} M(z)$$

In the time domain

$$m(k) = x(k) - \sum_{i=1}^n b_i m(k-i)$$

$$y(k) = \sum_{i=0}^n a_i m(k-i) \quad (12-3)$$

Equations (12-3) define the first direct (1D) structure as shown in Figure 12-1a. In Figure 12-1 time delay (z^{-1}) is represented by rectangular boxes; multipliers, by labeled arrows; adders, by circles and ellipses containing a plus (+); and signal distribution points, by dark dots at joining lines and dark bars. The 1D structure is called canonical because it possesses only n time-delay elements, the minimum number for an n th-order transfer function of (12-1). Note that this structure appeared in Figures 2-9 and 2-11. There the coefficient a_0 was set to zero and the structure was called the *control*, or *phase-variable*, *canonical form*.

Transpose Networks

The *transpose* structure of a digital filter structure is formed by reversing the signal flow in all branches of the block diagram [1]. Consequently, the summing junctions become signal distribution points, and vice versa. The input becomes the output, and

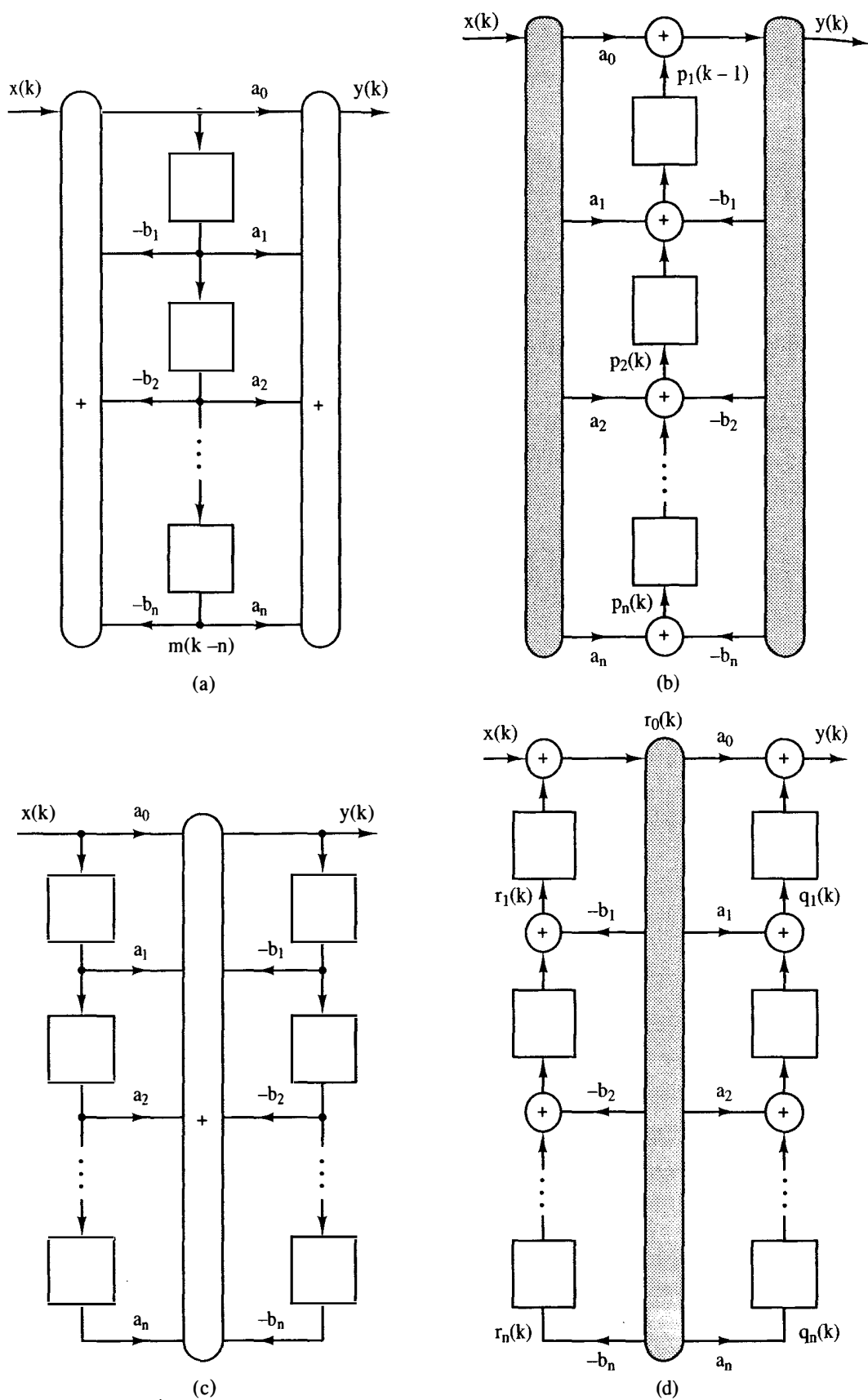


Figure 12-1 Direct structures: (a) 1D; (b) 2D; (c) 3D; (d) 4D. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 1, p. 24, Feb. 1981, © 1981 IEEE.)

vice versa. The transpose of a filter structure has the same transfer function as the original structure. Hence structures for digital filters exist as transpose pairs. Consequently, we may use these properties of structures to derive a second direct (2D) structure for (12-1).

Second Direct Structure

If we take the transpose of the 1D structure, we obtain the 2D structure shown in Figure 12-1b. It also implements (12-1), but it requires $n + 1$ difference equations (summing junctions), whereas the 1D structure required only 2. The 2D difference equations have the form

$$\begin{aligned} p_i(k) &= p_{i+1}(k-1) + a_i x(k) - b_i y(k), & i = 1, n-1 \\ p_n(k) &= a_n x(k) - b_n y(k) \\ y(k) &= a_0 x(k) + p_1(k-1) \end{aligned} \quad (12-4)$$

This structure is also canonical. We have also seen this structure earlier in Chapter 2. In Figure 2-10, with $a_0 = 0$, this structure was called the *observer canonical form*.

Third Direct Structure

Returning to (12-1) we may write

$$D(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^n a_i z^{-i}}{\sum_{i=0}^n b_i z^{-i}}$$

and thus

$$Y(z) \sum_{i=0}^n b_i z^{-i} = X(z) \sum_{i=0}^n a_i z^{-i}$$

Consequently,

$$Y(z) = \sum_{i=0}^n a_i z^{-i} X(z) - \sum_{i=1}^n b_i z^{-i} Y(z)$$

In the time domain

$$y(k) = \sum_{i=0}^n a_i x(k-i) - \sum_{i=1}^n b_i y(k-i) \quad (12-5)$$

This is the difference equation for the third direct (3D) structure, which is block diagrammed in Figure 12-1c. Notice that the structure has only one summing junction, but has $2n$ time-delay elements.

Fourth Direct Structure

The fourth direct (4D) structure is the transpose of the 3D structure, as shown in Figure 12-1d. This structure has only one signal distribution point, but has $2n$ difference equations, as expressed below:

$$\begin{aligned}
 r_0(k) &= x(k) + r_1(k-1) \\
 q_n(k) &= a_n r_0(k) \\
 r_n(k) &= -b_n r_0(k) \\
 q_i(k) &= a_i r_0(k) + q_{i+1}(k-1), \quad i = 1, n-1 \\
 r_i(k) &= -b_i r_0(k) + r_{i+1}(k-1) \\
 y(k) &= a_0 r_0(k) + q_1(k-1)
 \end{aligned} \tag{12-6}$$

Summary

Four direct structures for an n th-order digital filter have been derived. Table 12-1 summarizes their characteristics. Note that 1D and 2D conserve time-delay elements, while 1D and 3D conserve summing junctions. Conserving delay elements saves memory space in computer implementations, but memory is relatively inexpensive. Conserving summing junctions makes the control unit of a digital filter easier to design. Signal distribution usually has little impact, except in LSI applications, where routing is very important. All the direct structures suffer extreme coefficient sensitivity as n grows large. That is, a *small* change in a coefficient a_i or b_i , for n large, causes *large* changes in the zeros or poles of $D(z)$ of (12-1).

12.3 SECOND-ORDER MODULES

To avoid the coefficient sensitivity problems, the transfer function $D(z)$ of (12-1) is usually implemented as a cascade or parallel of second-order modules of the form

TABLE 12-1 PROPERTIES OF THE DIRECT STRUCTURES

	Structure			
	1D	2D	3D	4D
Time-delay elements	n	n	$2n$	$2n$
Multipliers	$2n + 1$	$2n + 1$	$2n + 1$	$2n + 1$
Summing junctions	2	$n + 1$	1	$2n$
Signal distribution points	$n + 1$	2	$2n$	1

Source: H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Table 1, p. 25, Feb. 1981, © 1981 IEEE.

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

The structure of these second-order modules can themselves be of the direct format of Figure 12-1. Figure 12-2 illustrates the 1D, 2D, 3D, and 4D structures for second-order modules.

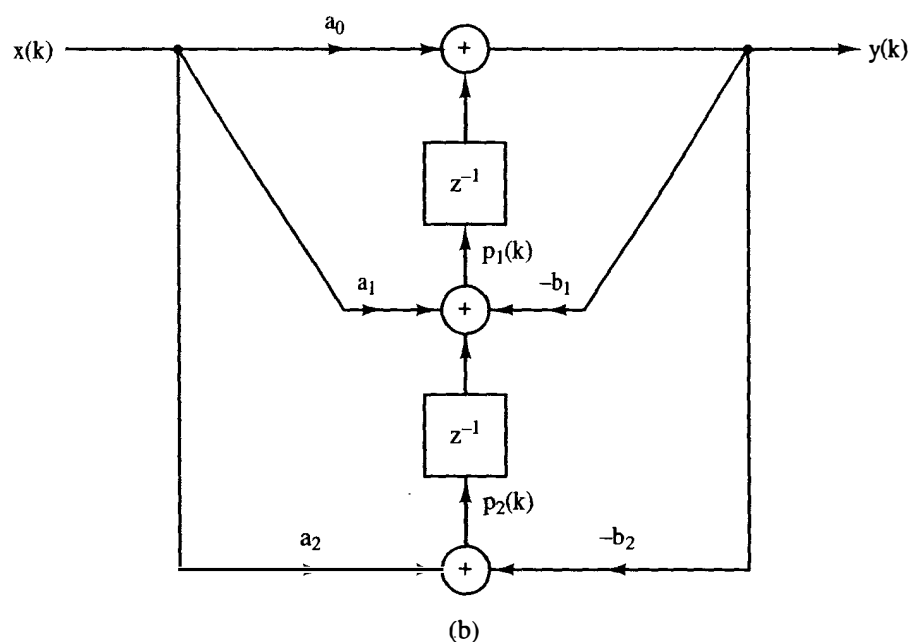
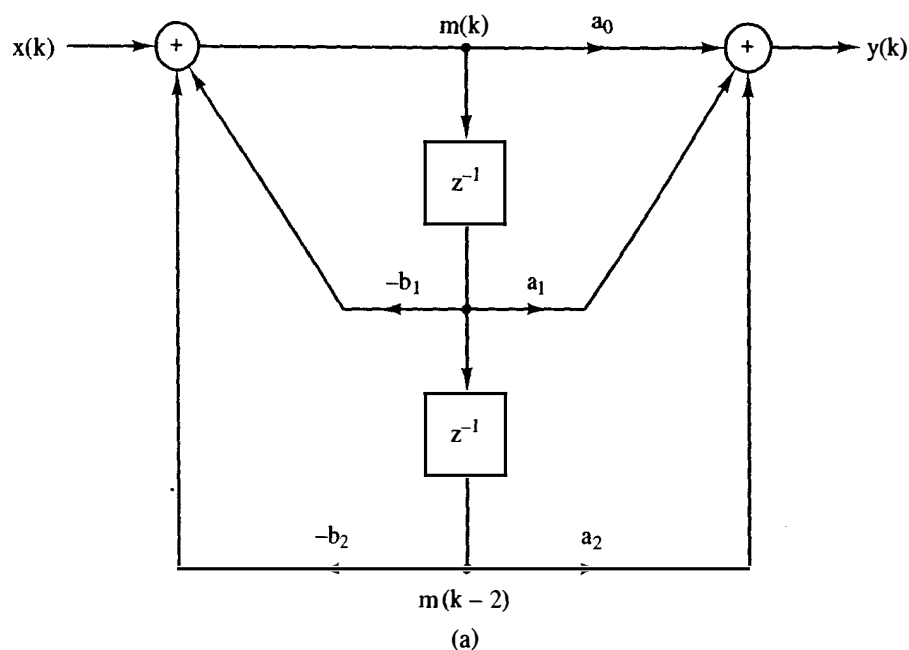
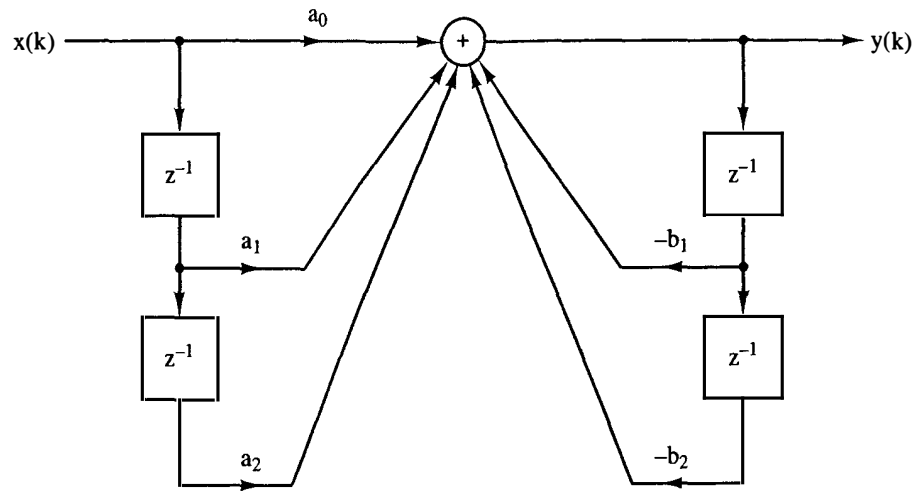
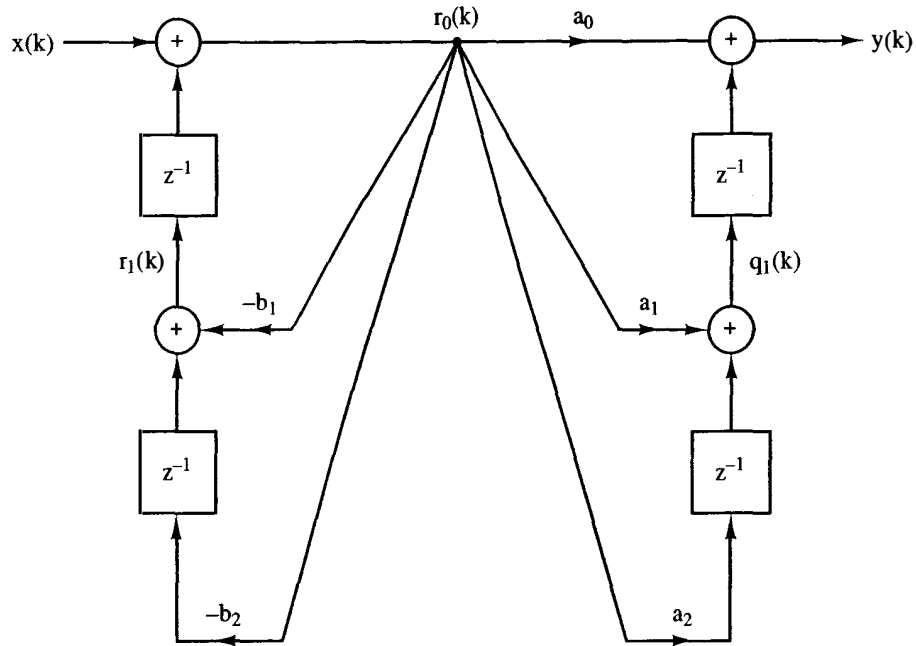


Figure 12-2 Direct second-order modules: (a) 1D; (b) 2D; (c) 3D; (d) 4D. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 2, p. 25, Feb. 1981, © 1981 IEEE.)



(c)



(d)

Figure 12-2 (continued)

The difference equations describing each structure are:

$$\begin{aligned} 1D: \quad m(k) &= x(k) - b_1 m(k-1) - b_2 m(k-2) \\ y(k) &= a_0 m(k) + a_1 m(k-1) + a_2 m(k-2) \end{aligned} \quad (12-7)$$

$$\begin{aligned} 2D: \quad y(k) &= a_0 x(k) + p_1(k-1) \\ p_1(k) &= a_1 x(k) - b_1 y(k) + p_2(k-1) \\ p_2(k) &= a_2 x(k) - b_2 y(k) \end{aligned} \quad (12-8)$$

$$\begin{aligned}
 \text{3D: } y(k) = & a_0 x(k) + a_1 x(k-1) + a_2 x(k-2) \\
 & - b_1 y(k-1) - b_2 y(k-2)
 \end{aligned}
 \quad (12-9)$$

$$\begin{aligned}
 \text{4D: } r_0(k) = & x(k) + r_1(k-1) \\
 y(k) = & a_0 r_0(k) + q_1(k-1) \\
 r_1(k) = & -b_1 r_0(k) - b_2 r_0(k-1) \\
 q_1(k) = & a_1 r_0(k) + a_2 r_0(k-1)
 \end{aligned}
 \quad (12-10)$$

The equations should be calculated in the proper order. For example, in (12-7), $m(k)$ *must* first be obtained; in (12-10), $r_0(k)$. In (12-8) and (12-9) $y(k)$ *should* be calculated first to minimize the calculation time delay between the input sample $x(kT)$ and output generation $y(kT + \tau_c)$, where τ_c represents the filter calculation delay. Ideally, $\tau_c = 0$, but since this is unattainable we minimize τ_c by ordering our calculations. Practically, then, if $T \gg \tau_c$ we can neglect τ_c .

Other structures for second-order modules are possible. The cross-coupled structure of Figure 12-3 has often appeared in the literature [2-4] for the complex-pole-pair case. Here we call it the 1X structure. The difference equations are:

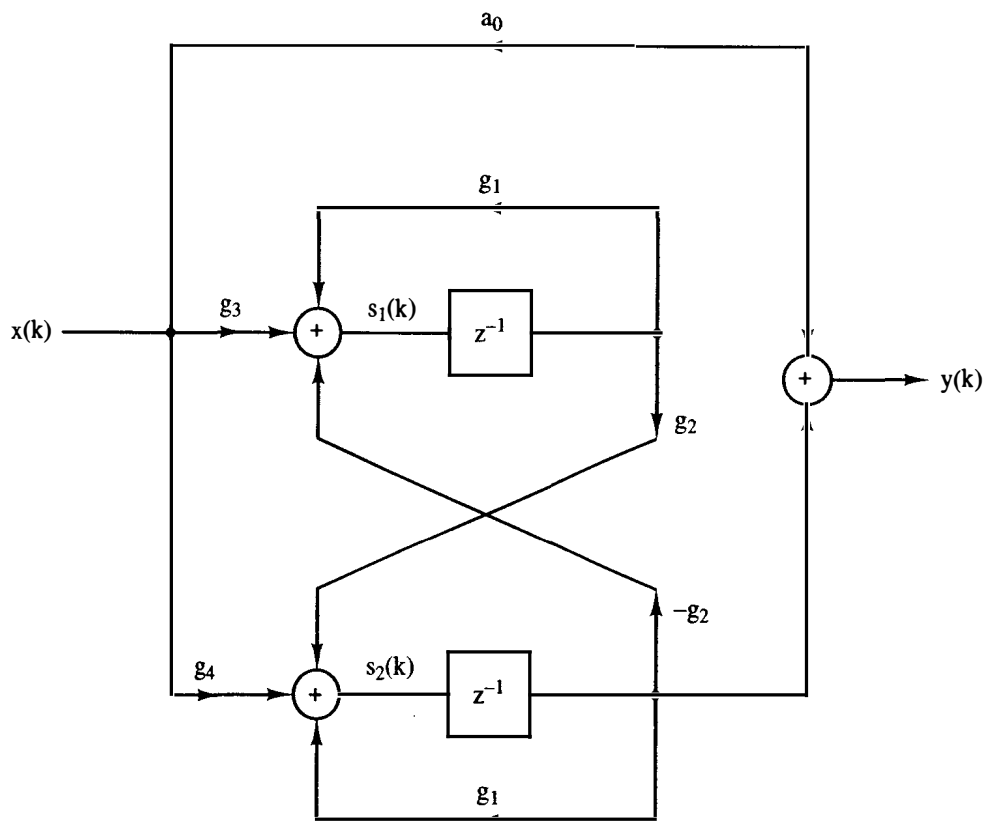


Figure 12-3 1X structure. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 3, p. 26, Feb. 1981, © 1981 IEEE.)

$$\begin{aligned}
 y(k) &= a_0 x(k) + s_2(k-1) \\
 s_1(k) &= g_1 s_1(k-1) - g_2 s_2(k-1) + g_3 x(k) \\
 s_2(k) &= g_1 s_2(k-1) + g_2 s_1(k-1) + g_4 x(k)
 \end{aligned}
 \tag{12-11}$$

where the g_i come from

$$D(z) = a_0 + \frac{A}{z+p} + \frac{A^*}{z+p^*}$$

and

$$\begin{aligned}
 g_1 &= -\text{Re}[p] \\
 g_2 &= -\text{Im}[p] \\
 g_3 &= 2 \text{Im}[A] \\
 g_4 &= 2 \text{Re}[A]
 \end{aligned}
 \tag{12-12}$$

Note that the 1X structure is canonical.

The transpose of the 1X structure is shown in Figure 12-4 and is termed the 2X structure. The difference equations are

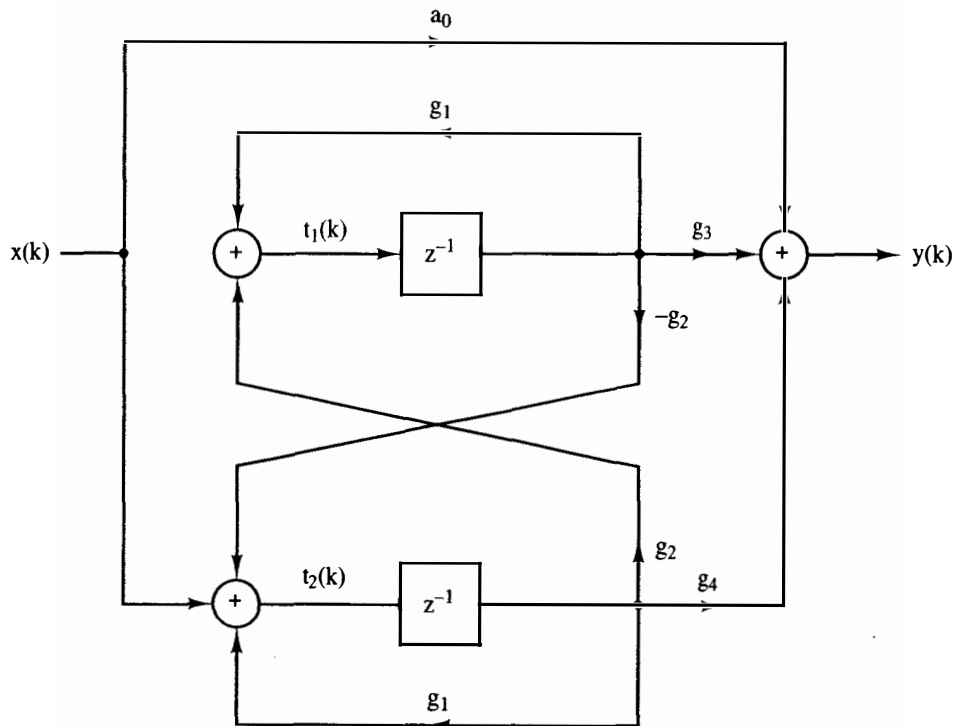


Figure 12-4 2X structure. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 4, p. 26, Feb. 1981, © 1981 IEEE.)

TABLE 12-2 SECOND-ORDER MODULES

	Structure					
	1D	2D	3D	4D	1X	2X
Time-delay elements	2	2	4	4	2	2
Multipliers	5	5	5	5	7	7
Summing junctions	2	3	1	4	3	3
Signal distribution points	3	2	4	1	3	3

Source: H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Table 2, p. 26, Feb. 1981, © 1981 IEEE.

$$\begin{aligned}
 y(k) &= a_0 x(k) + g_3 t_1(k-1) + g_4 t_2(k-1) \\
 t_1(k) &= g_1 t_1(k-1) + g_2 t_2(k-1) \\
 t_2(k) &= x(k) + g_1 t_2(k-1) - g_2 t_1(k-1)
 \end{aligned} \tag{12-13}$$

where g_i are defined in (12-12).

These six structures will be used for second-order modules in the remainder of this book. Table 12-2 summarizes the structures. The 1X and 2X structures require two more multipliers than are required by the direct structures.

12.4 CASCADE REALIZATION

To avoid coefficient-sensitivity problems, $D(z)$ of (12-1) may be implemented using a cascade of second-order modules. By factoring (12-1), we obtain

$$D(z) = \frac{\prod_{i=1}^m (\alpha_{i0} + \alpha_{i1} z^{-1} + \alpha_{i2} z^{-2})}{\prod_{i=1}^m (1 + \alpha_{i3} z^{-1} + \alpha_{i4} z^{-2})} \tag{12-14}$$

where m is the smallest integer greater than or equal to $n/2$. If the numerator and denominator factors are paired (the *pairing* problem) and the modules ordered in the cascade (the *ordering* problem), then

$$D(z) = \prod_{i=1}^m A_i(z)$$

where

$$A_i(z) = \frac{a_{i0} + \alpha_{i1} z^{-1} + \alpha_{i2} z^{-2}}{1 + \alpha_{i3} z^{-1} + \alpha_{i4} z^{-2}} \tag{12-15}$$

The pairing and ordering problems in cascaded second-order modules have been extensively studied in the literature [5-8]. In Chapter 14 we examine these problems

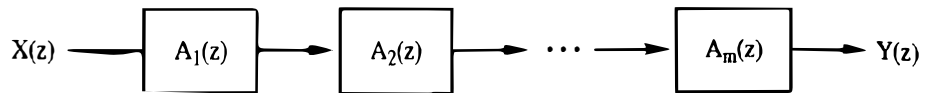
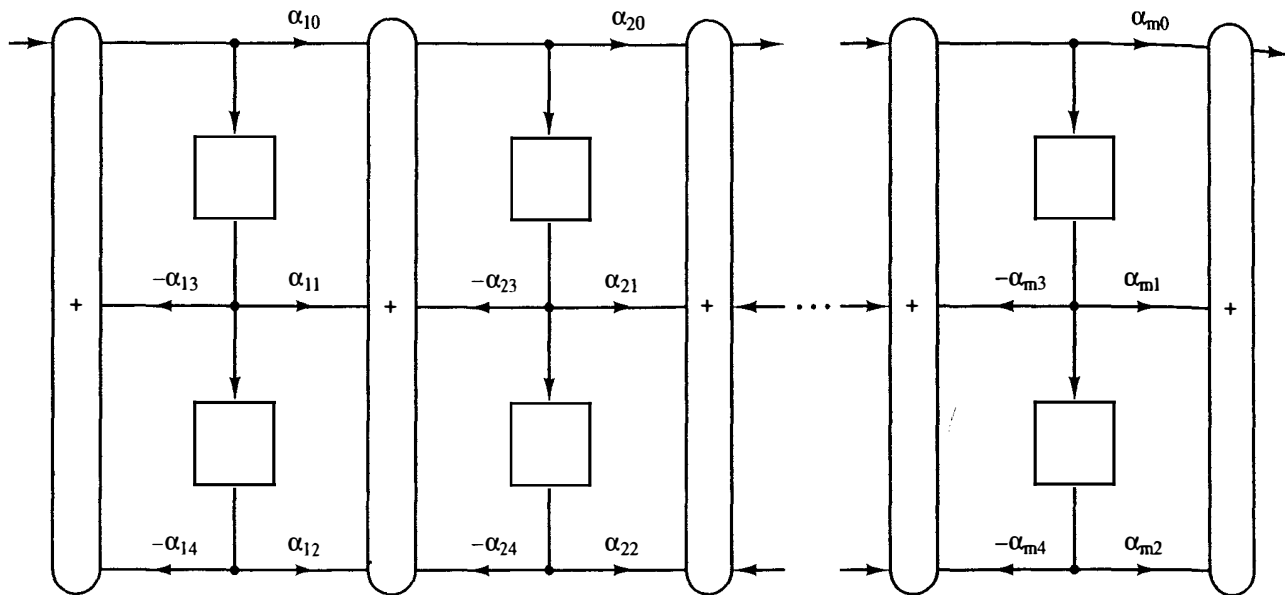
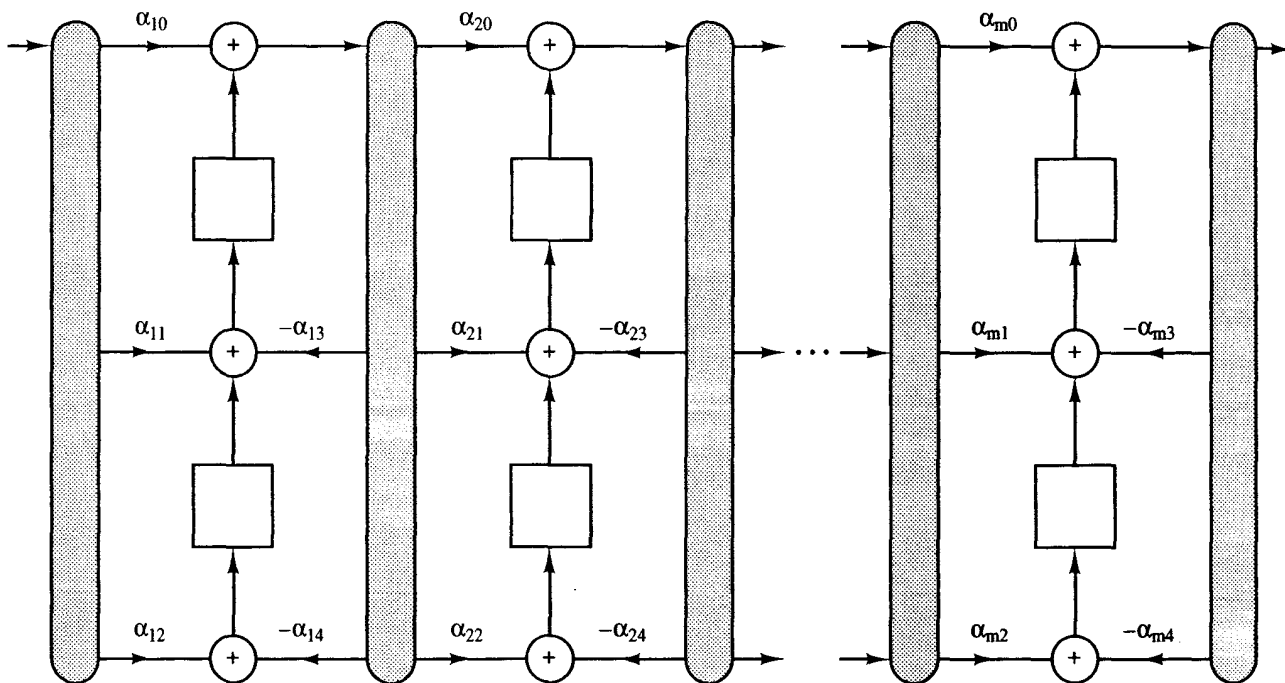


Figure 12-5 Cascaded second-order modules. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 5, p. 27, Feb. 1981, © 1981 IEEE.)



(a)



(b)

Figure 12-6 Cascade filter structures: (a) 1D; (b) 2D; (c) 3D; (d) 4D. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 6, p. 28, Feb. 1981, © 1981 IEEE.)

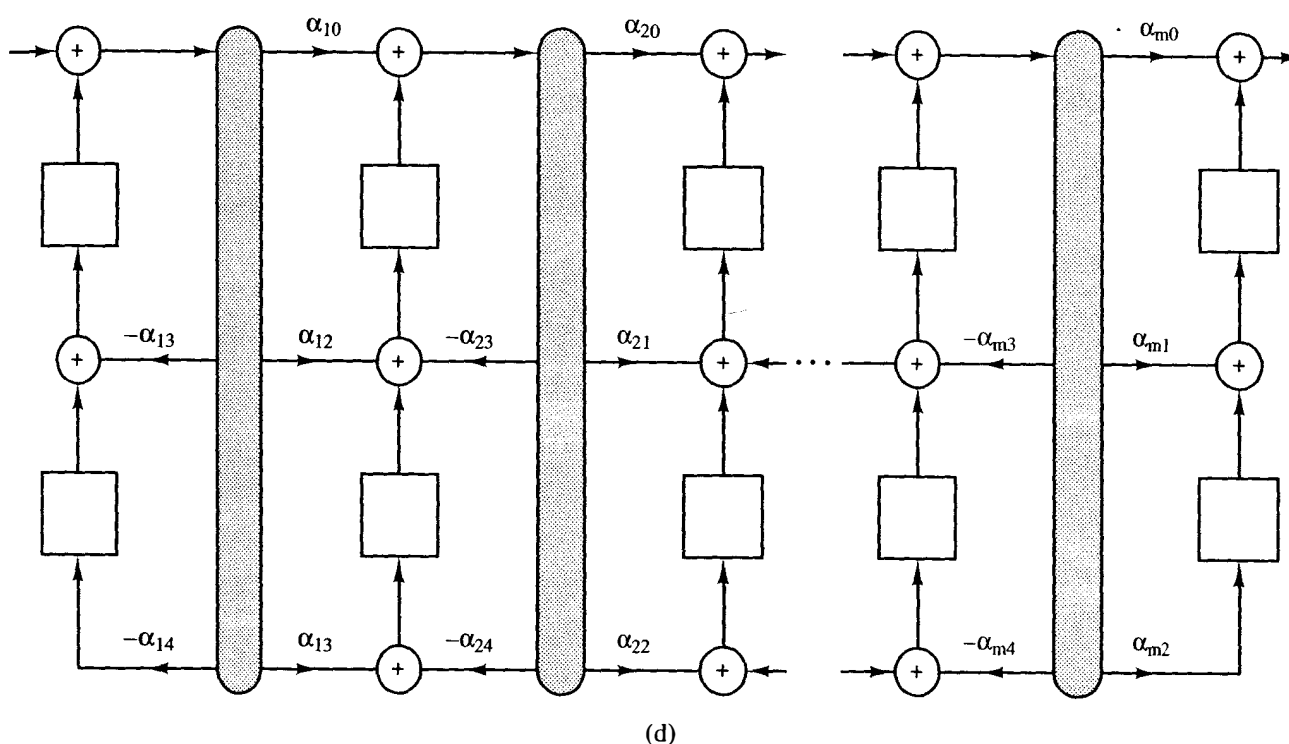
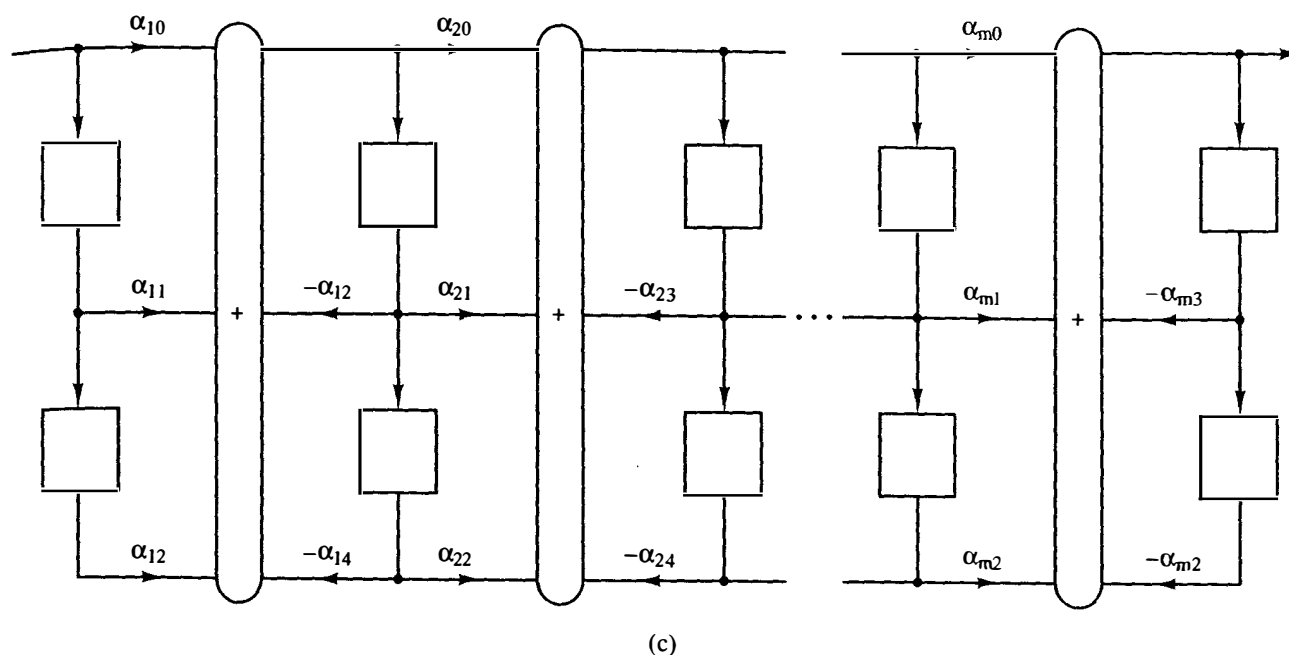


Figure 12-6 (continued)

more closely. Figure 12-5 illustrates the cascade of (12-15); the second-order modules may be implemented in the direct or cross-coupled structures. If the direct structures are used, the cascade diagrams of Figure 12-6 result. These cascade structures are compared in Table 12-3. If one contrasts Tables 12-1 and 12-3, we see that cascading 3D and 4D modules saves $n - 2$ delay elements in each case. Cascad-

TABLE 12-3 CASCADE STRUCTURES

	Structure ^a			
	1D	2D	3D	4D
Time-delay elements	$2m$ (n)	$2m$ (n)	$2m + 2$ ($2n - (n - 2)$)	$2m + 2$ ($2n - (n - 2)$)
Multipliers	$5m$ ($n + n + m$)	$5m$ ($n + n + m$)	$5m$ ($2n + m$)	$5m$ ($2n + m$)
Summing junctions	$m + 1$ ($2 + m - 1$)	$3m$ ($n + m$)	m	$3m + 1$ ($2n - (m - 1)$)
Signal distribution points	$3m$ ($n + m$)	$m + 1$ ($2 + m - 1$)	$3m + 1$ ($2n - (m - 1)$)	m

^a m is the smallest integer greater than or equal to $n/2$. The numbers in parentheses are for comparison with Table 12-1.

Source: H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Table 3, p. 27, Feb. 1981, © 1981 IEEE.

ing costs extra multipliers in every case. Cascaded direct modules require $m - 1$ extra summing junctions and signal distribution points over the direct structures.

12.5 PARALLEL REALIZATION

A second method to avoid the coefficient-sensitivity problems of (12-1) is to factor the denominator of $D(z)$ and to perform a partial-fraction expansion to obtain (for distinct poles)

$$D(z) = \beta_0 + \sum_{i=1}^m B_i(z)$$

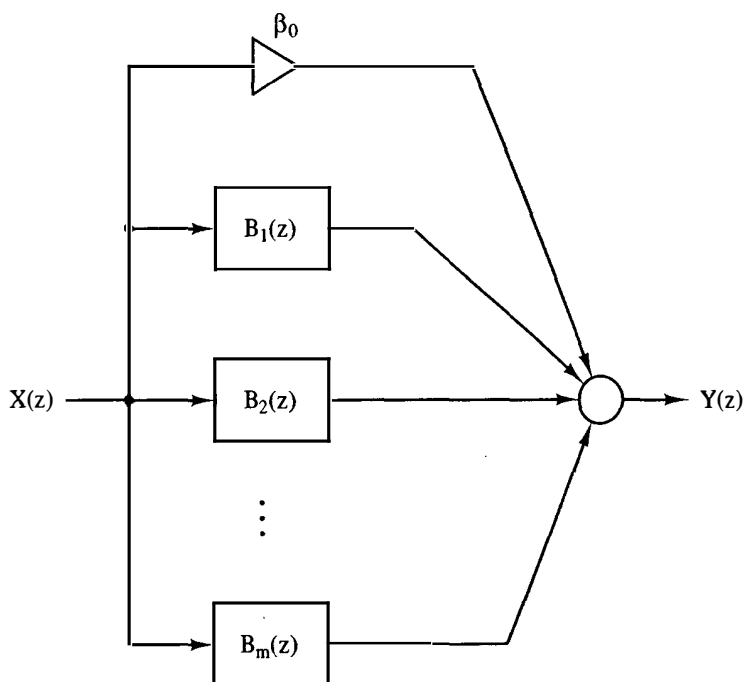


Figure 12-7 Parallel structure. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 7, p. 28, Feb. 1981, © 1981 IEEE.)

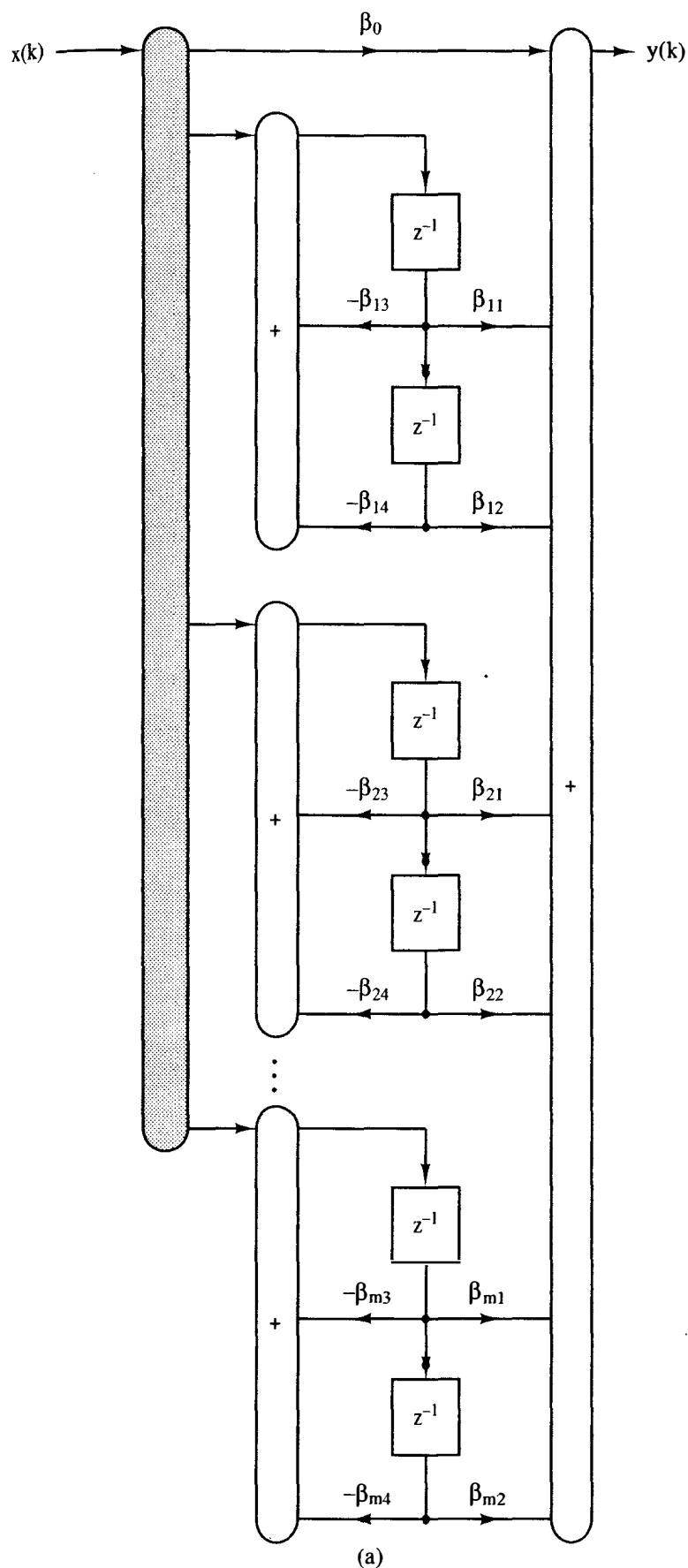
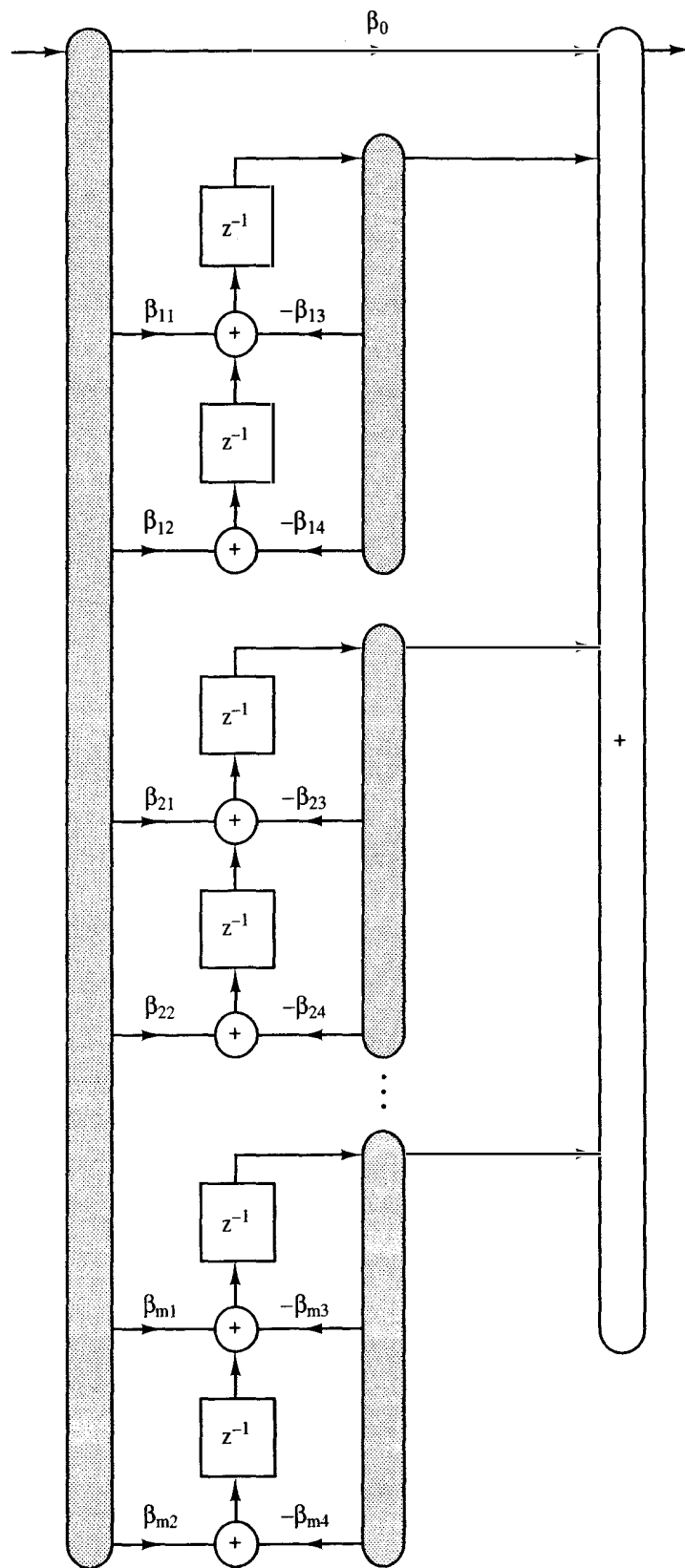
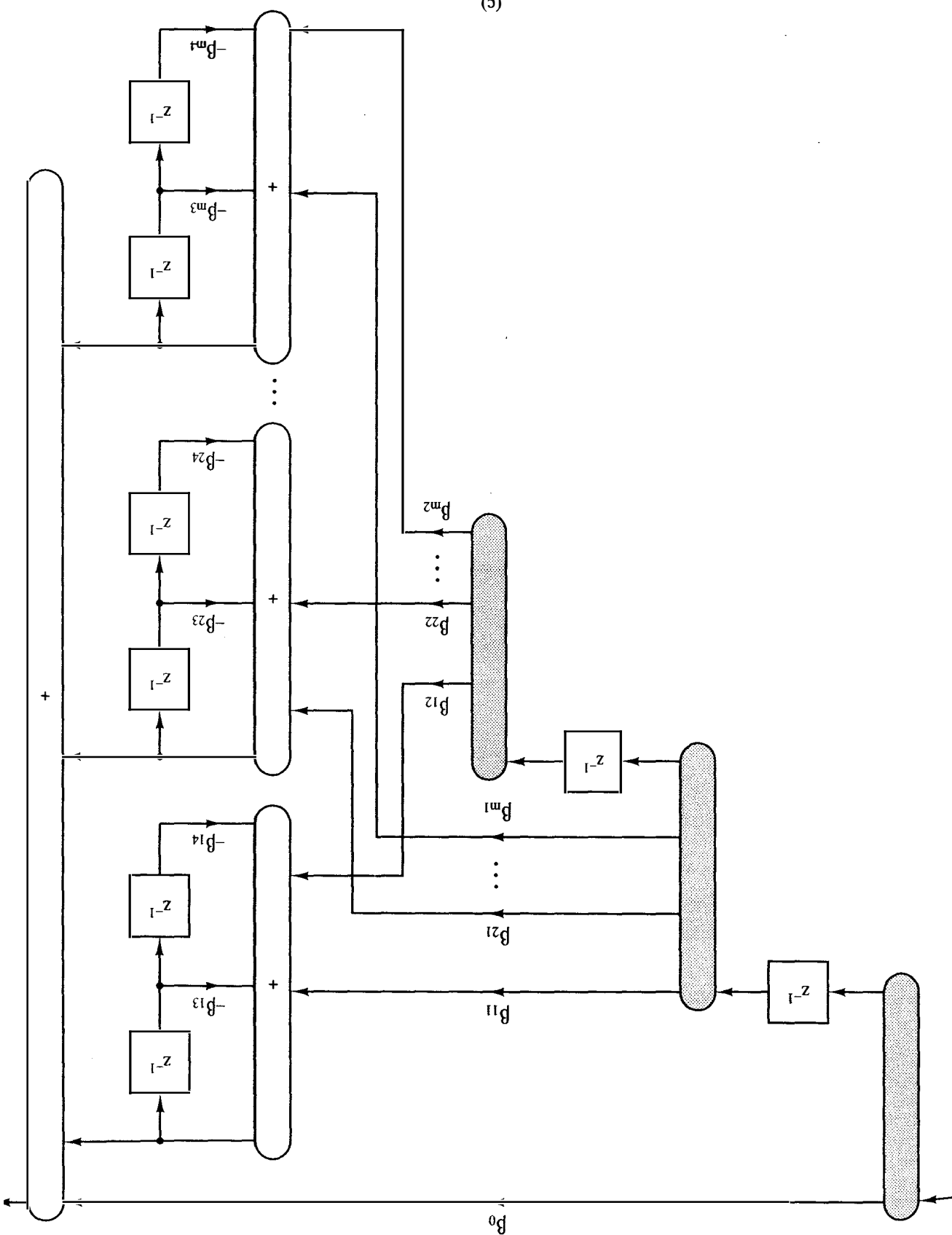


Figure 12-8 Paralleled second-order modules: (a) 1D; (b) 2D; (c) 3D; (d) 4D. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 8, pp. 28-30, Feb. 1981, © 1981 IEEE.)



(b)

Figure 12-8 (continued)



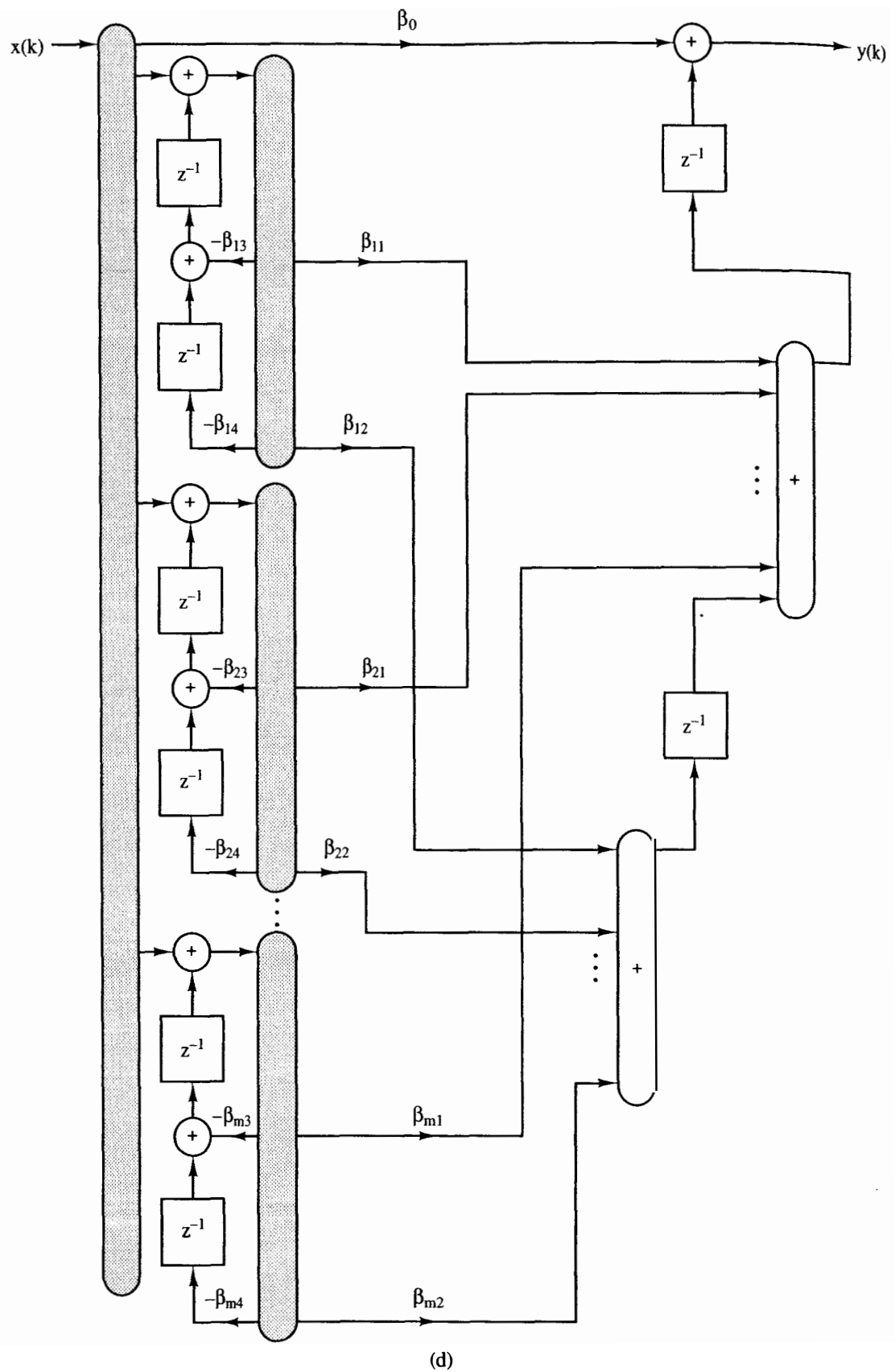


Figure 12-8 (continued)

TABLE 12-4 PARALLEL STRUCTURES

	Structure ^a			
	1D	2D	3D	4D
Time-delay elements	$2m$ (n)	$2m$ (n)	$2m + 2$ ($2n - (n - 2)$)	$2m + 2$ ($2n - (n - 2)$)
Multipliers	$4m + 1$ ($2n + 1$)	$4m + 1$ ($2n + 1$)	$4m + 1$ ($2n + 1$)	$4m + 1$ ($2n + 1$)
Summing junctions	$m + 1$ ($2 + m - 1$)	$2m + 1$ ($n + 1$)	$m + 1$ ($1 + m$)	$2m + 3$ ($2n - (n - 3)$)
Signal distribution points	$2m + 1$ ($n + 1$)	$m + 1$ ($2 + m - 1$)	$2m + 3$ ($2n - (n - 3)$)	$m + 1$ ($1 + m$)

^a m is the smallest integer greater than or equal to $n/2$. Numbers in parentheses are for comparison with Table 12-1.

Source: H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Table 4, p. 31, Feb. 1981, © 1981 IEEE.

where

$$B_i(z) = \frac{\beta_{i1}z^{-1} + \beta_{i2}z^{-2}}{1 + \beta_{i3}z^{-1} + \beta_{i4}z^{-2}} \quad (12-16)$$

Figure 12-7 depicts the parallel structure. Any of the six structures of Section 12.3 may be used to implement the blocks of Figure 12-7. If the direct structures are used, some element sharing may be accomplished, as was the case in the cascade implementation. Figure 12-8 indicates the direct parallel structures. Table 12-4 compares the characteristics of the parallel structures with those of Table 12-1. Note that the 3D and 4D parallel structures save $n - 2$ time delays over the direct realizations. The number of multipliers is the same as in Table 12-1. The 1D parallel structure requires an additional $m - 1$ summing junctions; the 2D, $m - 1$ signal distribution points. The 3D requires m additional summing junctions and $n - 3$ fewer signal distribution points (and alternately for the 4D parallel case).

12.6 PID CONTROLLERS

In Chapter 8 we presented the concept of digital control using proportion (K_P), integration (K_I), and differentiation (K_D) in (8-52):

This transfer function may be implemented as shown in Figure 12-9. Here the proportional, integral, and differential terms are implemented separately and summed at the output. An alternative method would be to find a second-order

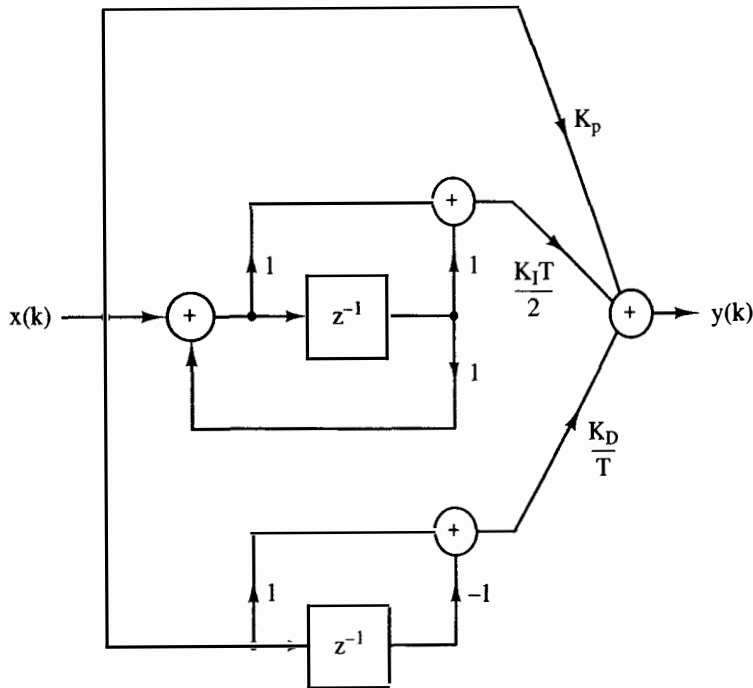


Figure 12-9 PID controller.

transfer function for (12-17) and use the direct structures presented earlier. Consider

$$\begin{aligned}
 D(z) &= \frac{K_P(z-1)(z) + (K_I T/2)(z+1)(z) + (K_D/T)(z-1)(z-1)}{(z-1)(z)} \\
 &= \frac{K_P(z^2 - z) + (K_I T/2)(z^2 + z) + (K_D/T)(z^2 - 2z + 1)}{z^2 - z} \\
 &= \frac{(K_P + K_I T/2 + K_D/T)z^2 + (-K_P + K_I T/2 - 2K_D/T)z + K_D/T}{z^2 - z} \\
 &= \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}
 \end{aligned}$$

where

$$\begin{aligned}
 a_0 &= K_P + \frac{K_I T}{2} = \frac{K_D}{T} \\
 a_1 &= -K_P + \frac{K_I T}{2} - \frac{2K_D}{T} \\
 a_2 &= \frac{K_D}{T} \\
 b_1 &= -1 \\
 b_2 &= 0
 \end{aligned} \tag{12-18}$$

Consequently, a PID controller can be implemented by any of the second-order, direct structures described earlier in the chapter.

12.7 LADDER REALIZATION

A third method for improving the coefficient sensitivity of the direct structures for (12-1) is to implement a ladder network [9-11]. If $D(z)$ in (12-1) is expressed as

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}}$$

where $a_n \neq 0$ and $b_n \neq 0$, then $D(z)$ can be expanded into continued-fraction form:

$$D(z) = A_0 + \frac{1}{B_1 z^{-1} + \frac{1}{A_1 + \frac{1}{B_2 z^{-1} + \frac{1}{\ddots + \frac{1}{B_n z^{-1} + \frac{1}{A_n}}}}}} \quad (12-19)$$

where A_i and B_i are real constants derived from a_i and b_i . If $a_n = 0$, then A_0 is zero. To implement (12-19) we must be able to implement

$$G_1(z) = \frac{1}{Bz^{-1} + T(z)} \quad (12-20)$$

$$G_2(z) = \frac{1}{A + T(z)}$$

These functions may be implemented as shown in Figure 12-10.

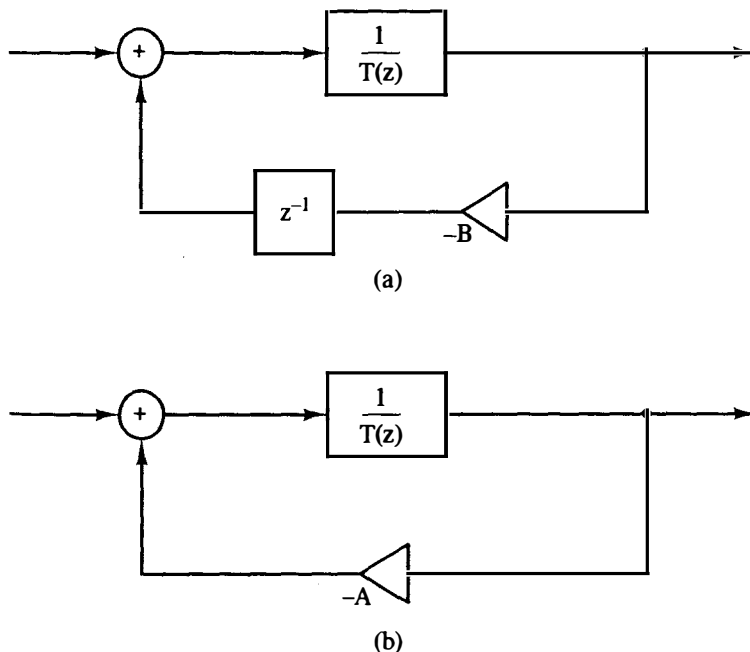


Figure 12-10 Continued fraction sections: (a) $G_1(z)$; (b) $G_2(z)$. [From S. K. Mitra and R. J. Sherwood, "Canonic Realizations of Digital Filters Using the Continued Fraction Expansion." *IEEE Trans. Audio Electroacoust.*, Vol. AU-20, No. 3, Aug. 1972. Part (a) from Fig. 6, p. 188; part (b) from Fig. 2, p. 186, © 1972 IEEE.]

The realization procedure is to first express (12-1) as

$$D(z) = A_0 + \frac{1}{B_1 z^{-1} + T_1(z)}$$

This part of the procedure is shown in Figure 12-11a. Next we express

$$T_1(z) = \frac{1}{A_1 + T_2(z)}$$

and

$$\frac{1}{T_1(z)} = A_1 + T_2(z)$$

This step is shown in Figure 12-11b. The third step is to express

$$T_2(z) = \frac{1}{B_2 z^{-1} + T_3(z)}$$

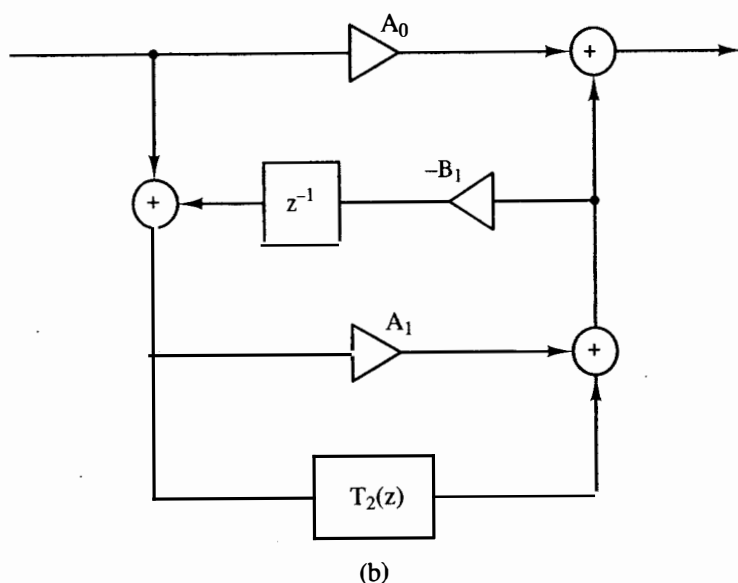
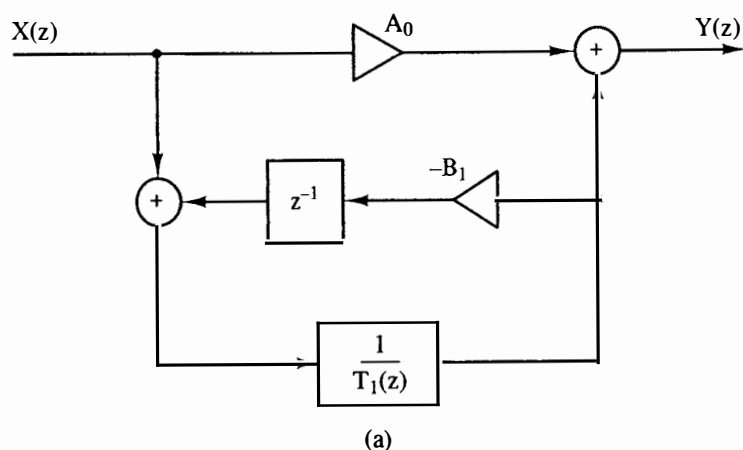


Figure 12-11 Ladder structure: (a) first step; (b) second step; (c) third step; (d) final step. [Part (d) from S. K. Mitra and R. J. Sherwood, "Canonic Realizations of Digital Filters Using the Continued Fraction Expansion." *IEEE Trans. Audio Electroacoust.*, Vol. AU-20, No. 3, Fig. 7, p. 188, Aug. 1972, © 1972 IEEE.]

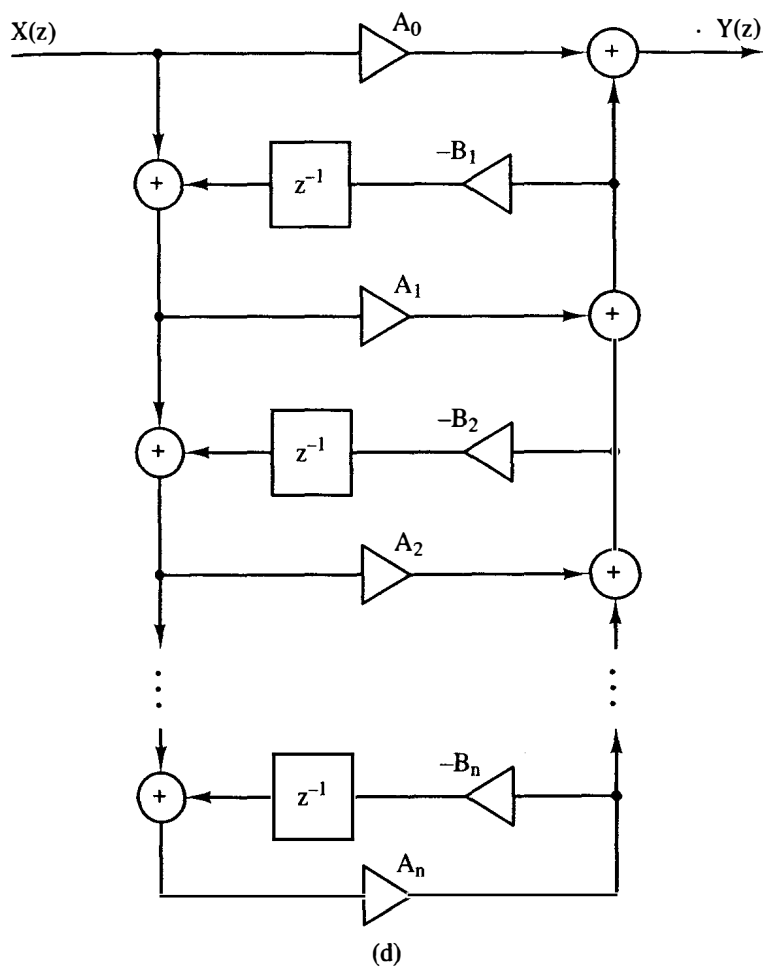
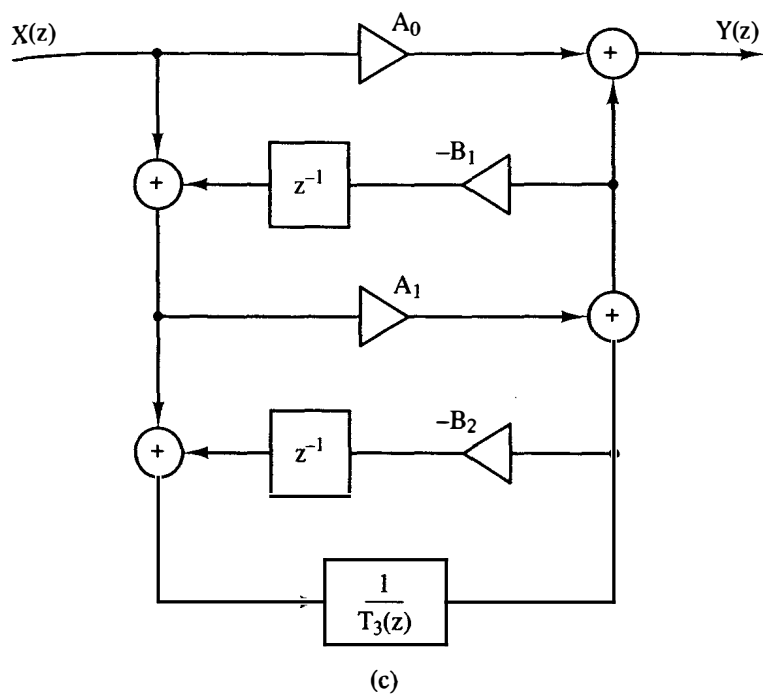


Figure 12-11 (continued)

which is essentially repeating step 1, as shown in Figure 12-11c. The process is repeated until we arrive at the ladder structure of Figure 12-11d.

There are many different digital ladder networks. However, we only show one to illustrate the principles involved. If real-time response is required, the ladder of Figure 12-11d is not recommended because all of the $2n$ difference equations must be calculated before the output is available.

The ladder structure is canonical because it requires n time-delay elements. It also requires $2n + 1$ multipliers and has $2n$ signal distribution points.

Example 12.1

Find a ladder realization of the following:

$$D(z) = \frac{11/128 + 106/128z^{-1} + 224/128z^{-2} + z^{-3}}{1/128 + 34/128z^{-1} + 160/128z^{-2} + z^{-3}}$$

In another form:

$$D(z) = \frac{128z^{-3} + 224z^{-2} + 106z^{-1} + 11}{128z^{-3} + 160z^{-2} + 34z^{-1} + 1}$$

First, divide the denominator into the numerator:

$$\begin{array}{r} 1 \\ 128 \ 160 \ 34 \ 1 \overline{) 128 \ 224 \ 106 \ 11} \\ \underline{128 \ 160 \ 34 \ 1} \\ 64 \ 72 \ 10 \end{array}$$

Therefore,

$$D(z) = 1 + \frac{1}{128z^{-3} + 160z^{-2} + 34z^{-1} + 1}$$

$$64z^{-2} + 72z^{-1} + 10$$

Again we divide the denominator into the numerator:

$$\begin{array}{r} 2z^{-1} \\ 64 \ 72 \ 10 \overline{) 128 \ 160 \ 34 \ 1} \\ \underline{128 \ 144 \ 20} \\ 16 \ 14 \ 1 \end{array}$$

Consequently,

$$D(z) = 1 + \frac{1}{2z^{-1} + \frac{1}{64z^{-2} + 72z^{-1} + 10}}$$

$$16z^{-2} + 14z^{-1} + 1$$

Repeating the procedure yields

$$\begin{array}{r} 4 \\ 16 \ 14 \ 1 \overline{) 64 \ 72 \ 10} \\ \underline{64 \ 56 \ 4} \\ 16 \ 6 \end{array}$$

The transfer function can then be written

$$D(z) = 1 + \frac{1}{2z^{-1} + \frac{1}{4 + \frac{1}{\frac{16z^{-2} + 14z^{-1} + 1}{16z^{-1} + 6}}}}$$

Again, repeating the division:

$$\begin{array}{r} z^{-1} \\ 16 \ 6 \) \ 16 \ 14 \ 1 \\ \underline{16 \ 6} \\ 8 \ 1 \end{array}$$

With each set, the order of the numerator or denominator is decreased by one.

$$D(z) = 1 + \frac{1}{2z^{-1} + \frac{1}{4 + \frac{1}{z^{-1} + \frac{16z^{-1} + 6}{8z^{-1} + 1}}}}$$

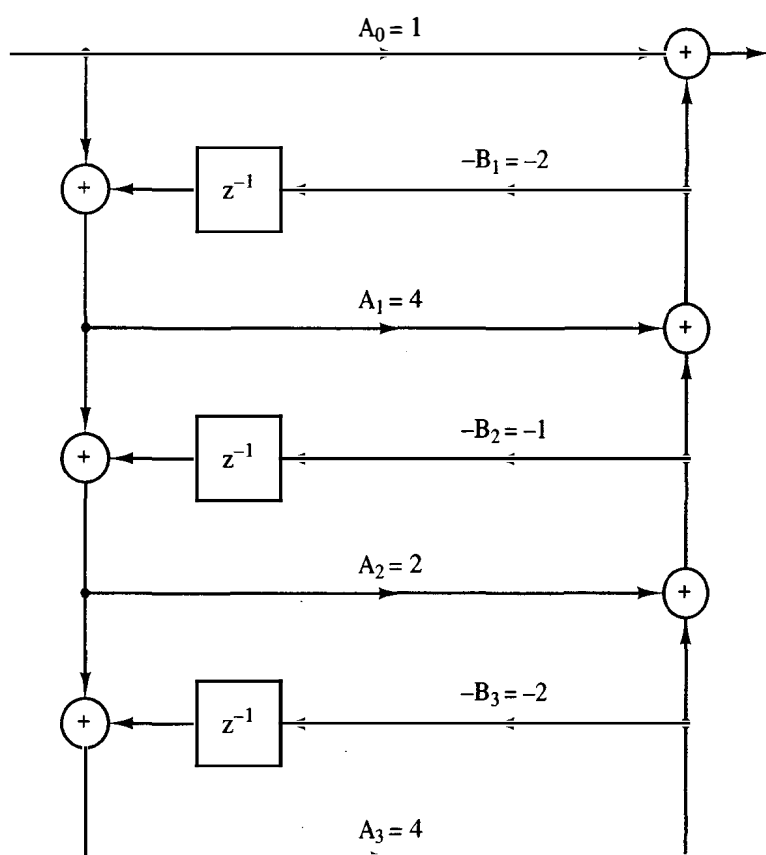


Figure 12-12 Example.

The final result is

$$D(z) = 1 + \frac{1}{2z^{-1} + \frac{1}{4 + \frac{1}{z^{-1} + \frac{1}{2 + \frac{1}{2z^{-1} + \frac{1}{4}}}}}}$$

The ladder structure is drawn in Figure 12-12.

12.8 OTHER STRUCTURES

Researchers in the field of signal processing have explored many avenues in the realization of digital filter structures. Fettweis [12] designed wave digital filters which exhibited many desirable properties. However, the implementation of the wave digital filter is more complex than the examples of this chapter. Other researchers have produced special structures to accomplish specific goals. Ali and Constantinides [13] designed low-coefficient sensitive structures. Fam and Barnes [14] concentrated on structure to eliminate limit cycles. Chang [15] emphasized low round-off noise. Abu-El-Haija et al. [16] have used a sampled-data transformation to represent an analog integrator, and have found filter structures for a digital incremental computer. Nishimura and Hirano [17] have tackled the problem of multiple feedback (leapfrog) digital filters.

12.9 SUMMARY

In this chapter we have examined various structures for digital filters. The most commonly used in digital control applications are the direct, cascade, and parallel structures. If the direct structure can implement a specified $D(z)$ within its frequency tolerances, cascading or paralleling second-order modules is unnecessary. However, if coefficient round-off causes the poles and zeros of $D(z)$ to move beyond tolerable limits, cascading or paralleling second-order modules is recommended for solving the problem.

REFERENCES

1. R. E. Crochiere and A. V. Oppenheim, "Analysis of Linear Digital Networks," *Proc. IEEE*, Vol. 63, pp. 581-595, Apr. 1975.
2. A. E. Vereshkin et al., "Two New Structures for the Implementation of a Discrete Transfer Function with Complex Poles," *Autom. Remote Control*, pp. 1416-1422, Sept. 1968.

3. H. T. Nagle, Jr., "Survey of Digital Filtering," Final Technical Report, NASA-CR-124166, Contract NAS8-20163, Oct. 1972, Auburn University, Auburn, AL, NASA Star CSDL09C, N73-20256.
4. L. B. Jackson, A. G. Lindgren, and Y. Kim, "Optimal Synthesis for Second-Order State-Space Structures for Digital Filters," *IEEE Trans. Circuits Syst.*, Vol. CAS-26, pp. 149-152, Mar. 1979.
5. L. B. Jackson, "Roundoff-Noise Analysis for Fixed-Point Digital Filters in Cascade or Parallel Form," *IEEE Trans. Audio Electroacoust.*, Vol. AU-18, pp. 107-122, June 1970.
6. S. Y. Hwang, "An Optimization of Cascade Fixed-Point Digital Filters," *IEEE Trans. Circuits Syst. (Letters)*, Vol. CAS-21, pp. 163-166, Jan. 1974.
7. W. S. Lee, "Optimization of Digital Filters for Low Roundoff Noise," *IEEE Trans. Circuits Syst.*, Vol. CAS-21, pp. 424-431, May 1974.
8. B. Liu and A. Peled, "Heuristic Optimization of the Cascade Realization of Fixed-Point Digital Filters," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-23, pp. 464-473, Oct. 1975.
9. S. K. Mitra and R. J. Sherwood, "Canonic Realizations of Digital Filters Using the Continued Fraction Expansion," *IEEE Trans. Audio Electroacoust.*, Vol. AU-20, No. 3, pp. 185-194, Aug. 1972.
10. S. K. Mitra and R. J. Sherwood, "Digital Ladder Networks," *IEEE Trans. Audio Electroacoust.*, Vol. AU-21, pp. 30-36, Feb. 1973.
11. L. T. Bruton, "Low-Sensitivity Digital Ladder Filters," *IEEE Trans. Circuits Syst.*, Vol. CAS-22, pp. 168-176, Mar. 1975.
12. A. Fettweis, "Some Principles of Designing Digital Filters Imitating Classical Filter Structure," *IEEE Trans. Circuit Theory*, pp. 314-316, Mar. 1971.
13. A. M. Ali and A. G. Constantinides, "Design of Low Sensitivity and Complexity Digital Filter Structures," *1978 Eur. Conf. Circuit Theory Des.*, Lausanne, Switzerland, Sept. 1978, pp. 335-339.
14. A. T. Fam and C. W. Barnes, "Nonminimal Realizations of Fixed-Point Digital Filters That Are Free of All Finite Work-Length Limit Cycles," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-27, pp. 149-153, Apr. 1979.
15. T. L. Chang, "A Low Roundoff Noise Digital Filter Structure," *ISCAS '78*, pp. 1004-1008.
16. A. I. Abu-El-Haija, K. Shenoi, and A. M. Peterson, "Digital Filter Structures Having Low Errors and Simple Hardware Implementation," *IEEE Trans. Circuits Syst.*, Vol. CAS-25, pp. 593-599, Aug. 1978.
17. S. Nishimura and K. Hirano, "Realizations of Digital Filters Using Generalized Multiple-Feedback Structure," *ISCAS '78*, pp. 284-288.

PROBLEMS

12-1. Examine the structure of Figure P12-1. If $\alpha_1 = 0$, find α_0 , α_2 , and α_3 such that

$$D(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

Note: $\alpha_i = f_i(a_0, a_1, a_2, b_1, b_2)$.

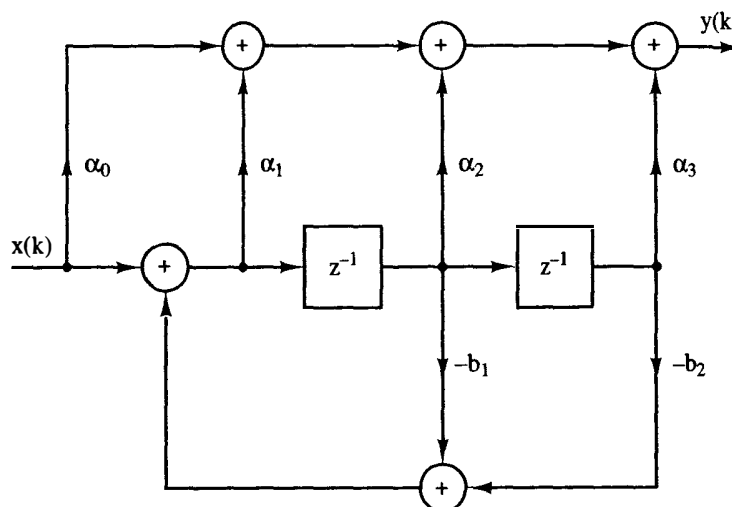


Figure P12-1

12-2. Repeat Problem 12-1 if $\alpha_2 = 0$.

12-3. Repeat Problem 12-1 if $\alpha_3 = 0$.

12-4. Examine Figure P12-4. Find $\alpha_0, \alpha_1, \alpha_2$, such that

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

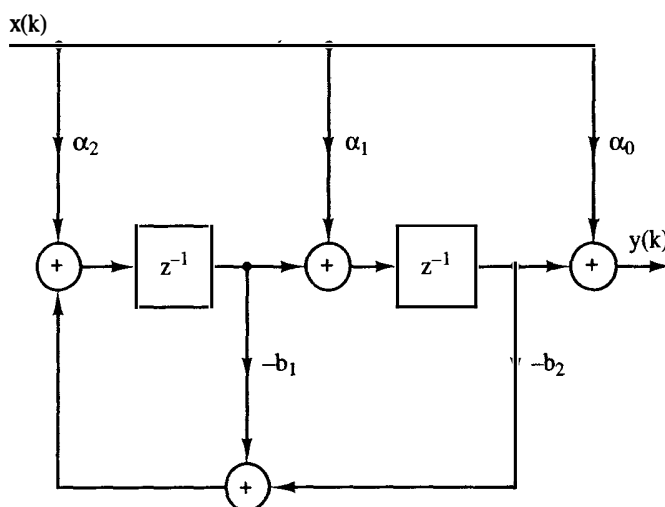


Figure P12-4

12-5. Consider the PID controller of Problem 11-9. Find a direct structure realization.

12-6. Consider the PID controller of Problem 11-10. Find a direct structure realization.

12-7. Can you find a 1X realization for the PID controllers of Problems 12-5 and 12-6?

Computer Implementation of Digital Filters

13.1 INTRODUCTION

In Chapter 12 we examined various structures for realizing digital filters. Each structure may be described by a unique set of difference equations. In this chapter we explore the implementation of these difference equations by computer. First we illustrate assembly language programming techniques for the popular Intel 80×86 series of processors. The techniques we present are widely applicable to the many different commercially available microprocessors and digital signal processing chip sets. Then we close the chapter by describing digital filter implementations using LabVIEW, a graphical programming language from National Instruments.

13.2 THE INTEL 80×86 [1]

The Intel 8086 was first introduced in 1978. It is a first-generation 16-bit microprocessor and represented a major advance in processing capability when it was introduced. Intel followed the 8086 with the 8088 version a year later, with the principal difference being the physical memory organization. The 8088 has an 8-bit data bus. The 8087 is a family of arithmetic coprocessors designed to run in parallel with the 8086/8088. The 8087 performs arithmetic operations about 100 times faster than the basic 8086/8088. Hardware implementation of the multiply operation is highly desirable for high-speed digital controller implementations.

The 80186/80188 and the 80286 are enhanced versions of the 8086/8088. The 80186 has a great deal more internal hardware than the original 8086: a clock

generator, programmable timers, DMA controller, chip select unit, and interrupt controller. The 80286 is a more advanced version of the 8086 that is designed for multiuser environments, possessing the capability of addressing 16 megabytes of physical memory. Unlike the 80186, the 80286 does not contain programmable timers and interrupt controller. Instead, it contains a memory management unit and several additional instructions.

In 1986 Intel introduced the 80386, a full 32-bit version of the 8086. It features memory management, multitasking, virtual memory, and software protection. The memory address space is expanded to 4 gigabytes. The 80486 is essentially the 80386 and its 80387 math coprocessor, all in one chip. In 1989 came the 80486, which uses a RISC internal structure to execute many instructions in just one clock cycle to achieve significant speed increases over the 80386. The 80486 has over 1 million transistors on a single chip. In 1993 the Pentium processor appeared, with over 3.1 million transistors and a clock frequency of 66 MHz [2].

Since all these processors are "upward" compatible, assembly programs for the 8086 are easily used on the more advanced models. Consequently, the programs described herein are for the 8086 processor and therefore should execute on any of the 80×86 family of machines.

Register Architecture

The register architecture of the Intel 8086 is shown in Figure 13-1. These registers are divided into four groups, including the (1) general register file, (2) pointer and index register file, (3) segment register file, and (4) instruction pointer (IP) and flag register.

Arithmetic and logical operations are supported by all of the general, pointer, and index registers, although typically only the general registers (AX, BX, CX, DX) are used. The accumulator (AX) is the most efficient in most operations and is explicitly implied in a number of operations, such as multiply and divide. Any of the general registers may be addressed as words or bytes for source and destination operands. In addition to arithmetic and logical functions, a number of string operations have been provided which use the general registers as indicated by their mnemonics in Figure 13-1. Thus, for string operations, CX is assumed to be a counter, BX a base address register, and AX and DX are assumed to contain data.

The capabilities of the 8086 are further enhanced by its many modes of memory access. The CPU can directly address 1 megabyte of memory. Each 20-bit address is computed from two components, as shown in Figure 13-2. The contents of a segment register is multiplied by 16 and then added to a 16-bit offset to determine the physical address in memory. Thus each of the four segment registers effectively points to one 64-kilobyte block of memory, resulting in separate 64K blocks for code (via CS), data (DS), and stack (SS), with the extra segment (ES) typically used as a second data-segment register. The offset of each address is computed using various combinations of constants and base or index register contents.

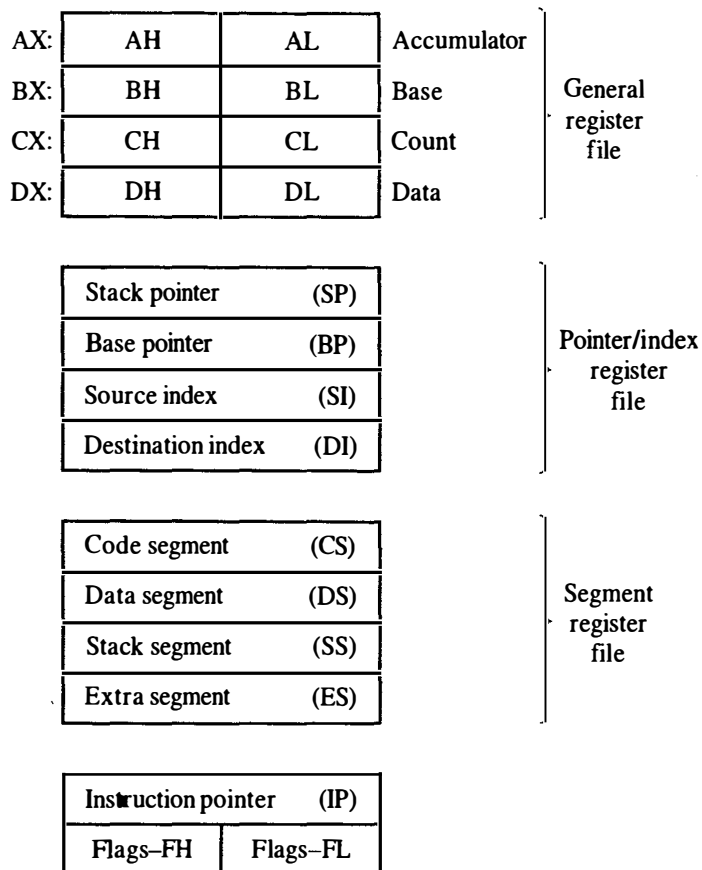


Figure 13-1 8086 register architecture.

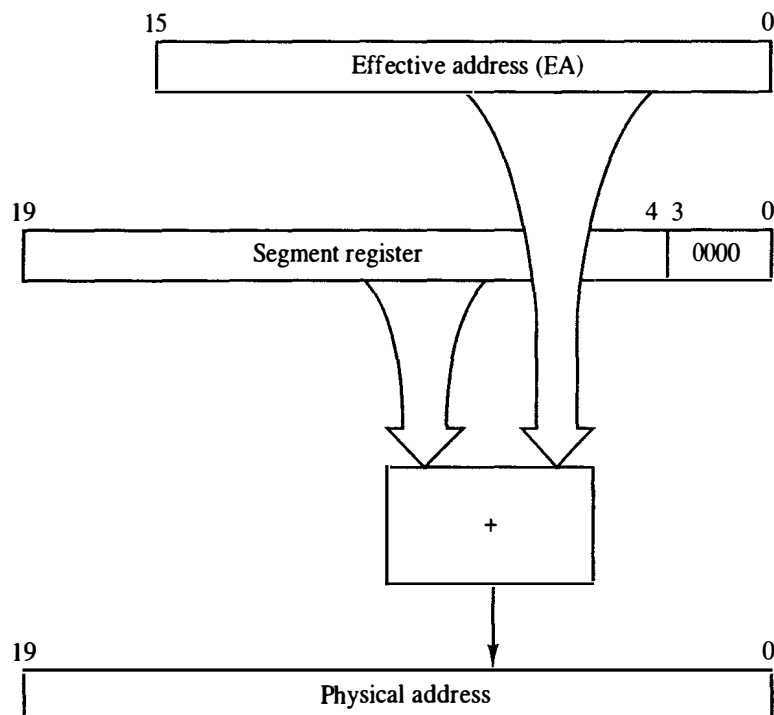


Figure 13-2 Physical address computation in the 8086.

In the case of instruction and stack operations, the segment and offset registers are explicitly implied; that is,

$$\text{instruction PA} = (\text{CS}) + (\text{IP})$$

$$\text{stack PA} = (\text{SS}) + (\text{SP})$$

where PA is the 20-bit physical address in memory. Operand addresses can be computed via a number of registers. The various operand addressing modes are summarized below.

1. Register: operand in {AX, BX, CX, DX, BP, SP, SI, DI}
2. Immediate: operand in instruction
3. Direct: $\text{PA} = (\text{DS}) + \text{DISP}$
4. Indirect:

$$\text{PA} = (\text{DS}) + (\text{BX}) + \text{DISP}$$

$$\text{PA} = (\text{SS}) + (\text{BP}) + \text{DISP}$$

$$\text{PA} = (\text{DS}) + \{(\text{SI}) \text{ or } (\text{DI})\} + \text{DISP}$$

$$\text{PA} = (\text{SS}) + (\text{BX}) + \{(\text{SI}) \text{ or } (\text{DI})\} + \text{DISP}$$

The segment register may be changed via a segment-override prefix. The displacement (DISP) may be 0, 8, or 16 bits.

8086 Instruction Set

The complete instruction set for the 8086 is available in Ref. 1. Here we describe a few of its more useful features for digital filter programming:

1. IMUL/MUL: Signed and unsigned multiply (byte or word). Operand 1 is in AL or AX, and operand 2 is accessed via the addressing modes. The result is left in AX (byte op's) or DX, AX (word op's).
2. Loop control
 - a. LOOP ADDR: The contents of CX are decremented by 1 and control is transferred to ADDR until (CX) = 0.
 - b. LOOPZ (LOOPNZ): Similar to loop, but control is only transferred if ZF (zero flag) is set (not set).
3. String operations
 - a. MOVS: Block move (byte or word operands) from $\langle(\text{SI})\rangle$ to $\langle(\text{DI})\rangle$, followed by an increment/decrement of both SI and DI.
 - b. REP MOVS: Block move with CX decremented after each move and the move repeated until (CX) = 0.
 - c. LODS/STOS: String load/store of 1 byte or word to/from AL to AX with

(SI) as the operand address. SI is incremented/decremented after the transfer.

- d. CLD/STD: Set direction flag to zero (auto decrement mode) or 1 (auto increment mode) for string operations.
- 4. CBW/CWD: Convert byte (word) to word (double-word) by extending the sign of AL (AX) through AH (DX).
- 5. XCHG: Exchange contents of registers or register and memory.

The use of these 8086 instructions simplifies the implementation of the iterative calculations required in digital filtering as well as providing a minimum phase lag from filter input to the filter output.

13.3 IMPLEMENTING SECOND-ORDER MODULES [3]

In Chapter 12 we examined six structures for second-order modules. Here we examine their implementation on the Intel 8086. Consider Figure 13-3. All programs for second-order modules may be modeled by this flowchart, which represents the processing required during one time interval ($kT < t < kT + T$). For example, consider the 1D structure of (12-7). If we precalculate (during the time interval $kT - T \leq t < kT$)

$$\begin{aligned} T_1 &= -b_1 m(k-1) - b_2 m(k-2) \\ T_2 &= a_1 m(k-1) + a_2 m(k-2) \end{aligned} \quad (13-1)$$

Then (during the time interval $kT \leq t < kT + T$) we may rapidly calculate the output $y(k)$ upon receipt of the input $x(k)$ as follows:

$$\begin{aligned} m(k) &= x(k) + T_1 \\ m(k) &= a_0 m(k) + T_2 \end{aligned} \quad (13-2)$$

This completes the processing of a 1D module so that no postprocessing is required.

Equations (12-7) through (12-12) are reorganized below into the routines of Figure 13-3.

1D Structure:

$$\begin{aligned} \text{OUTP_1D: } m(k) &= x(k) + T_1 \\ y(k) &= a_0 m(k) + T_2 \\ \text{POST_1D: } &\text{none} \\ \text{PRE_1D: } T_1 &= -b_1 m(k-1) - b_2 m(k-2) \\ T_2 &= a_1 m(k-1) + a_2 m(k-2) \end{aligned} \quad (13-3)$$

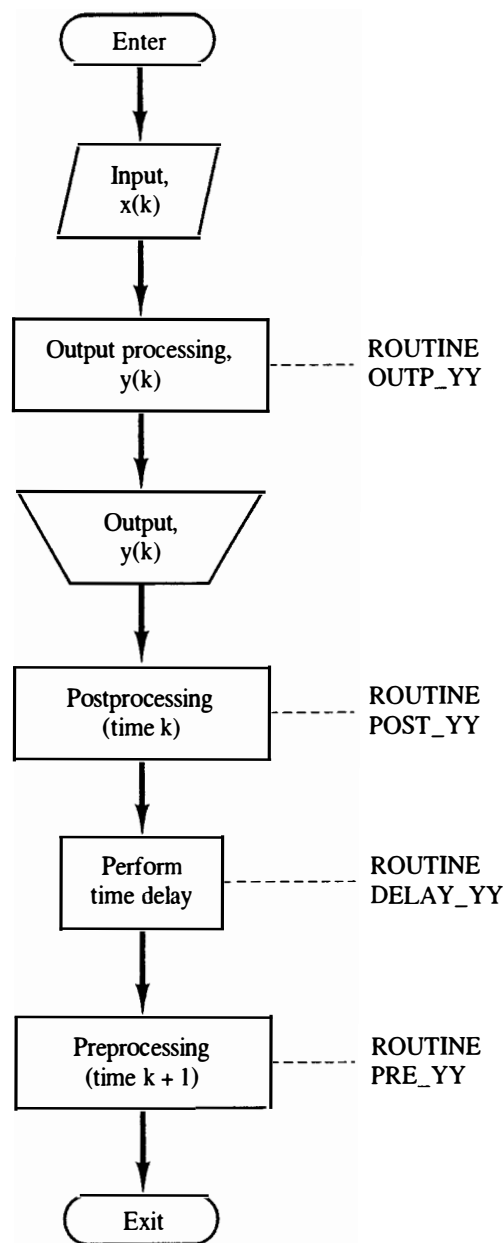


Figure 13-3 General second-order module. YY = 1D, 2D, 3D, 4D, 1X, 2X. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 9, p. 31, Feb. 1981, © 1981 IEEE.)

2D Structure:

$$\text{OUTP_2D: } y(k) = a_0x(k) + p_1(k-1)$$

$$\text{POST_2D: } p_1(k) = a_1x(k) - b_1y(k) + p_2(k-1)$$

$$p_2(k) = a_2x(k) - b_2y(k)$$

$$\text{PRE_2D: none}$$

(13-4)

3D Structure:

$$\begin{aligned}
 \text{OUTP_3D: } y(k) &= a_0 x(k) + T_3 \\
 \text{POST_3D: } &\text{none} \\
 \text{PRE_3D: } T_3 &= a_1 x(k-1) + a_2 x(k-2) \\
 &\quad - b_1 y(k-1) - b_2 y(k-2)
 \end{aligned} \tag{13-5}$$

4D Structure:

$$\begin{aligned}
 \text{OUTP_4D: } r_0(k) &= x(k) + r_1(k-1) \\
 y(k) &= a_0 r_0(k) + q_1(k-1) \\
 \text{POST_4D: } r_1(k) &= -b_1 r_0(k) - b_2 r_0(k-1) \\
 q_1(k) &= a_1 r_0(k) + a_2 r_0(k-1) \\
 \text{PRE_4D: } &\text{none}
 \end{aligned} \tag{13-6}$$

1X Structure:

$$\begin{aligned}
 \text{OUTP_1X: } y(k) &= a_0 x(k) + s_2(k-1) \\
 \text{POST_1X: } s_1(k) &= g_1 s_1(k-1) - g_2 s_2(k-1) + g_3 x(k) \\
 s_2(k) &= g_1 s_2(k-1) + g_2 s_1(k-1) + g_4 x(k) \\
 \text{PRE_1X: } &\text{none}
 \end{aligned} \tag{13-7}$$

2X Structure:

$$\begin{aligned}
 \text{OUTP_2X: } y(k) &= a_0 x(k) + T_4 \\
 \text{POST_2X: } t_1(k) &= g_1 t_1(k-1) + g_2 t_2(k-1) \\
 t_2(k) &= x(k) + g_1 t_2(k-1) - g_2 t_1(k-1) \\
 \text{PRE_2X: } T_4 &= g_3 t_1(k-1) + g_4 t_2(k-1)
 \end{aligned} \tag{13-8}$$

The 1X and 2X structures require a pair of complex poles so that

$$D(z) = a_0 + \frac{A}{z+p} + \frac{A^*}{z+p^*} \tag{13-9}$$

and

$$\begin{aligned}
 g_1 &= -\text{Re}[p] & g_3 &= 2 \text{Im}[A] \\
 g_2 &= -\text{Im}[p] & g_4 &= 2 \text{Re}[A]
 \end{aligned}$$

Figure 13-4 illustrates the calling sequences for the filter structure modules of Figure 13-3. Intel 8086 assembly language programs for the 1D structure appear in

INPUT:	No parameters passed. Returns sample $x(k)$ in AX.
OUTP_YY:	Pass $x(k)$ in AX. Put number of cascaded stages in CX. Returns output $y(k)$ in AX.
DELAY_YY:	Pass number of modules in CX to perform time delay.
PRE_YY:	Pass number of modules in CX for preprocessing.
POST_YY:	Pass number of modules in CX for postprocessing.
YY = 1D, 2D, 3D, 4D, 1X, or 2X.	

Figure 13-4 Calling sequences for filter subroutine modules.

Figure 13-5. Programs for the other second-order structures are listed in Appendix VI. To operate these routines correctly it is important that the structure coefficients be entered in a specific format described below.

Note that the coefficient values are bounded by

$$\begin{aligned} 0 &\leq |a_0, a_2, b_2, g_1, g_2| \leq 1 \\ 0 &\leq |a_1, b_1, g_3, g_4| \leq 2 \end{aligned} \quad (13-10)$$

Since the two's-complement number system is used in the Intel 8086, all numbers in the machine must be represented by

$$N = (SM_{14} \ M_{13} \ \cdots \ M_1 \ M_0)_{2\text{cns}} \quad (13-11)$$

Hence

$$-2^{15} \leq N \leq 2^{15} - 1 \quad (13-12)$$

If we consider all numbers to be scaled such that

$$N = (S \cdot M_{14} \ M_{13} \ \cdots \ M_1 \ M_0)_{2\text{cns}} \quad (13-13)$$

Thus

$$-1 \leq N \leq 1 - 2^{-15} \quad (13-14)$$

Consequently, coefficients in the range

$$1 \leq |N| < 2 \quad (13-15)$$

cannot be represented. Therefore, all coefficients will be stored as *half* their actual value, $\text{VALUE_STORED} = [\text{Value} * 2^{14} + .5]$ and a left shift (multiply by 2) operation will be performed in each routine to compensate for this change. The symbol $[x]$ means the largest integer less than x .

```

;OUTP_1D:  M0 = X + T1  :  Y = A0*M0 + T2
;X PASSED IN AX, Y RETURNED IN AX
;LOOP COUNT IN CX
OUTP_1D:  MOV     SI, #0      ; LOOP INDEX
          LEA     DI, M0      ; MEMORY POINTER
OLP_1D:   ADD     AX, T1[SI]  ; M0
          STOW                    ; STORE
          IMUL    A0[SI]      ; A0*M0/4 IN DX
          SAL     DX, 2       ; A0*M0
          ADD     DX, T2[SI]  ; Y
          MOV     AX, DX      ; Y IN AX
          ADD     SI, #2      ; MOVE INDEX
          LOOP    OLP_1D     ; LOOP BACK
          RET

;
;
;
;DELAY_1D:  M2 = M1, M1 = M0 FOR TIME-DELAY
;LOOP COUNT IN CX
DELAY_1D:  LEA     DI, T1-2   ; POINT TO M2
          LEA     SI, M2-2   ; POINT TO M1
          STD                    ; SET AUTODECREMENT
          SAL     CX, 1       ; DOUBLE LOOP COUNT
          REP
          MOVW                    ; BLOCK MOVE
          CLD
          RET

;
;
;
;PRE_1D:   T1 = -B1*M1 - B2*M2
          T2 = A1*M1 + A2*M2
;LOOP COUNT IN CX
PRE_1D:   LEA     SI, A1      ; POINT TO COEFS
          MOV     DI, #0      ; INDEX
PLP_1D:   LODW                    ; A1/2
          IMUL    M1[DI]      ; M1*A1/4 IN DX
          MOV     BX, DX      ; SAVE
          LODW                    ; A2/2
          IMUL    M2[DI]      ; M2*A2/4 IN DX
          ADD     BX, DX      ; T2/4
          SAL     BX, 2       ; T2
          MOV     T2[DI], BX  ; STORE T2
          LODW                    ; B1/2
          IMUL    M1[DI]      ; M1*B1/4 IN DX
          MOV     BX, DX      ; SAVE
          LODW                    ; B2/2
    
```

Figure 13-5 1D module subroutines.

```

IMUL    M2[DI]      ; M2*B2/4
ADD     BX, DX      ; -T1/4
SAL     BX, 2       ; -T1
NEG     BX          ; T1
MOV     T1[DI], BX  ; STORE T1
ADD     DI, #2      ; MOVE INDEX
LOOP    PLP_1D      ; LOOP BACK
RET

;
;
;
; POST_1D:  NOT USED IN A 1D MODULE
POST_1D:  RET
;
;
;
; 1D DATA STORAGE FOR N STAGES
A0:      DW  A10, A20, ..., AN0      ; A0 COEFS
A1:      DW  A11, A12, A13, A14      ; STAGE 1 A1, A2, B1, B2
          DW  A21, A22, A23, A24      ; STAGE 2
.
.
          DW  AN1, AN2, AN3, AN4      ; STAGE N
M0:      DW  NDUP (0)                ; M(k)
M1:      DW  NDUP (0)                ; M(k-1)
M2:      DW  NDUP(0)                 ; M(k-2)
T1:      DW  NDUP(0)                 ; TEMP STORAGE
T2:      DW  NDUP(0)                 ; TEMP STORAGE

```

Figure 13-5 (continued)

Example 13.1

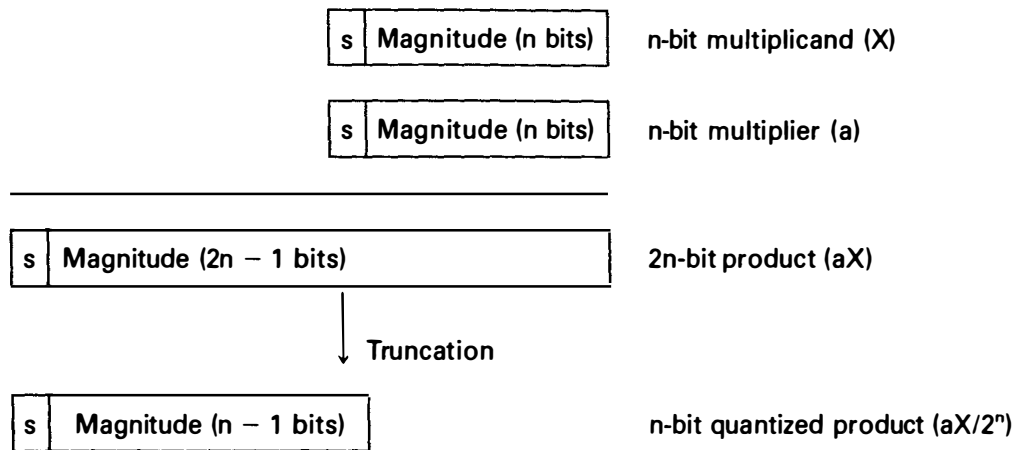
Suppose that coefficient $S_0 = 0.4383164$ is to be stored in the Intel 8086.

$$\begin{aligned}
 \text{VALUE_STORED} &= [S_0 * 2^{14} + .5] \\
 &= [.4383164 * 16384 + .5] \\
 &= [7181.8759] = 7181
 \end{aligned} \tag{13-16}$$

Next let us consider multiplication in the two's-complement number system. Figure 13-6a illustrates the problem. An n -bit multiplicand (perhaps a signal variable) is multiplied by an n -bit multiplier (perhaps a filter coefficient) and the product has $2n$ bits. This product may be used as another multiplicand in a later multiplication, so it is quantized (truncated here) back to n bits. The effect of this quantization is examined in great detail in Chapter 14.

Suppose that the multiplicand is X and the coefficient is a ; then

$$\begin{array}{r}
 X \\
 \times a \\
 \hline
 aX
 \end{array}$$



```

LEA S1, A    ;COEF POINTER
LODW         ;A/2 LOADED
IMUL X       ;AX/4 IN DX
SAL DX, 2    ;AX IN DX
    
```

(b)

Figure 13-6 Two's-complement multiplication: (a) wordlength variation; (b) code sequence.

and the product is quantized $Q[\cdot]$:

$$Q[aX] = [aX/2^n] \quad (13-17)$$

But from (13-11) and (13-13) we may see that the computer hardware actually handles the integers of (13-11) so that in hardware

$$\frac{X * 2^{n-1} \cdot a * 2^{n-1}}{aX * 2^{2n-2}}$$

is quantized

$$Q[aX] = [aX * 2^{2n-2}/2^n] = [aX * 2^{n-2}] \quad (13-18)$$

Consequently, the product must be multiplied by 2 (shifted one place left) so that the final truncated term is

$$Q[aX] = [aX * 2^{n-1}] \quad (13-19)$$

Earlier we explained that coefficients are actually stored as half-values; hence the computer actually performs the following operations:

1. Load the coefficient a into AX:

$$AX = a * 2^{n-2}$$

2. Multiply by the variable X:

$$\begin{aligned} DX, AX &= (a * 2^{n-2}) * (X * 2^{n-1}) \\ &= aX * 2^{2n-3} \\ &= (aX/4) * 2^{2n-1} \end{aligned}$$

The product is now in the DX, AX register.

3. Shift DX left two places (quantize to 16 bits and multiply by 4):

$$\begin{aligned} DX &= [aX/4 * 2^{2n-1}/2^n] * 4 \\ &= [(aX/4) * 2^{n-1}] * 4 \\ &\doteq aX * 2^{n-1} \end{aligned}$$

This operation left justifies the register DX and fills in two zeros in the least significant bits. The DX register now contains the truncated, properly scaled result. The instruction sequence is listed in Figure 13-11b. This sequence appears frequently in the computer programs of this chapter.

4. On computers with double register shifting, one would perform the double left shift first:

$$\text{double register} = (aX/4 * 2^{2n-1}) * 4 = aX * 2^{2n-1}$$

and then truncate to the n most significant bits.

$$\begin{aligned} \text{Single register} &= [(aX * 2^{2n-1})/2^n] \\ &= [aX * 2^{n-1}] \end{aligned}$$

which is more accurate than above.

Example 13.2

Consider multiplying

$$X = .7562867$$

and

$$a = .4383164$$

From (13-16),

$$[a * 2^{14}] = 7181$$

and from (13-13),

$$[X * 2^{15}] = 24782$$

After multiplication

$$aX * 2^{29} = 177959542$$

On the Intel 8086 executing the program of Figure 13-6, truncation is accomplished before shifting left; hence

$$(aX * 2^{29})/2^{16} = 177959542/2^{16}$$

$$[aX * 2^{13}] = [2715.4471] = 2715$$

Finally, the shift left operation multiplies by 4:

$$\begin{aligned} [aX * 2^{13}] * 2^2 &= 2715 * 4 \\ &= 10860 \end{aligned} \quad (13-20)$$

Hence

$$\begin{aligned} aX &\doteq 10860/2^{15} \\ &\doteq .3314209 \end{aligned}$$

The actual answer is 0.3314929.

On a computer with double register shifting, first we multiply by 4:

$$(aX * 2^{29}) * 4 = 177959542 * 4$$

$$aX * 2^{31} = 711838168$$

and then divide by 2^{16} and truncate:

$$\begin{aligned} (aX * 2^{31})/2^{16} &= 711838168/2^{16} \\ [aX * 2^{15}] &= [10861.788] \\ &= 10861 \end{aligned} \quad (13-21)$$

and

$$\begin{aligned} aX &= 10861/2^{15} \\ &= 0.3314514 \end{aligned}$$

which is more accurate. The Intel 8086 can perform this more accurate computation by replacing the SAL DX, 2 instruction with the instruction sequence:

```
RCL  AX, 1
RCL  DX, 1
RCL  AX, 1
RCL  DX, 1
```

This replacement will slow down the filter's operating speed slightly.

13.4 PARALLEL IMPLEMENTATION OF HIGHER-ORDER FILTERS

For implementing higher-order filters, we express $D(z)$ in the form of (12-16):

$$D(z) = \beta_0 + \sum_{i=1}^m B_i(z) \quad (13-22)$$

where

$$B_i(z) = \frac{\beta_{i1} z^{-1} + \beta_{i2} z^{-2}}{1 + \beta_{i3} z^{-1} + \beta_{i4} z^{-2}}$$

Figure 13-7 depicts the storage allocation for all modules. Coefficients and variable storage are grouped for more efficient addressing.

Example 13.3

Consider the implementation of a fourth-order digital filter. This is a rate filter used in the vehicle control portion of the space shuttle [4]:

$$D(z) = \frac{1 + 0.390244z^{-1} - 1.24247z^{-2} + 0.344333z^{-3} + 0.977044z^{-4}}{1 - 3.02828z^{-1} + 3.53682z^{-2} - 1.88867z^{-3} + 0.397506z^{-4}} \quad (13-23)$$

In the parallel form,

$$D(z) = \beta_0 + \beta_1 \frac{\beta_{11} z^{-1} + \beta_{12} z^{-2}}{1 + \beta_{13} z^{-1} + \beta_{14} z^{-2}} + \beta_2 \frac{\beta_{21} z^{-1} + \beta_{22} z^{-2}}{1 + \beta_{23} z^{-1} + \beta_{24} z^{-2}} \quad (13-24)$$

where

$$\begin{aligned} \beta_0 &= 1.0, & \beta_1 &= -8.0, & \beta_2 &= -4.0 \\ \beta_{11} &= 1.263998 & \beta_{21} &= 1.673365 \\ \beta_{12} &= -1.747506 & \beta_{22} &= -1.569220 \\ \beta_{13} &= -1.823002 & \beta_{23} &= -1.205277 \\ \beta_{14} &= 0.895895 & \beta_{24} &= 0.443701 \end{aligned}$$

Using the 1D modules, the structure of Figure 13-8 is obtained. Note that β_1 and β_2 represent shifting operations in the computer. Using the 1D routines of Figure 13-5, the parallel implementation of Figure 13-9 is obtained.

13.5 CASCADE IMPLEMENTATION OF HIGHER-ORDER FILTERS

For implementing filters as a cascade of second-order modules, we may write $D(z)$ in the form of (12-15):

$$D(z) = \prod_{i=1}^m A_i(z) \quad (13-25)$$

```
COEF:  DW                ;FOR ALL MODULES, HALF VALUES
                                ;ROM OR RAM
;
VAR:    DW    0           ;FOR ALL MODULES
                                ;RAM
;
TEMP:   DW    0           ;FOR ALL MODULES
                                ;RAM
```

Figure 13-7 Higher-order storage allocation.

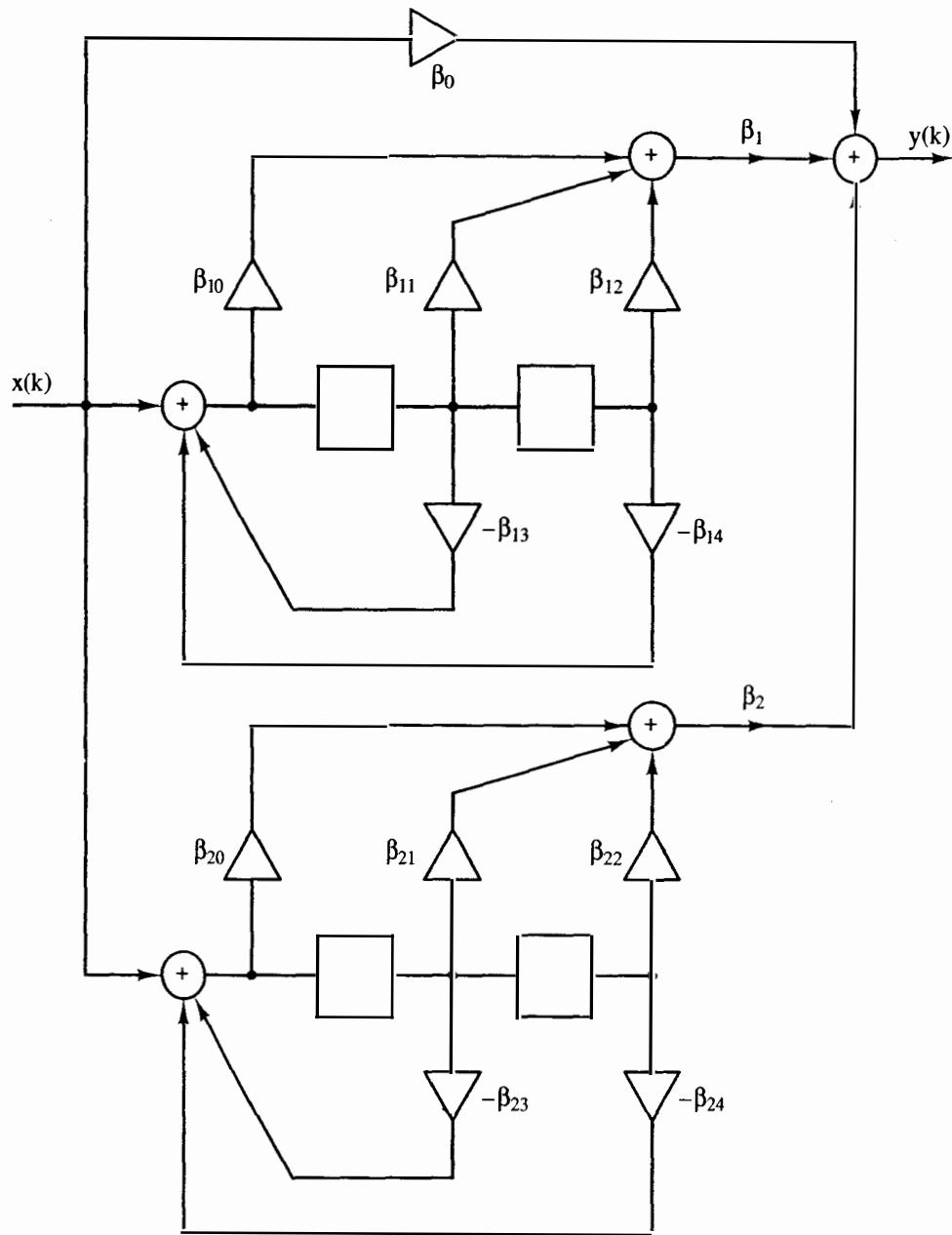


Figure 13-8 Fourth-order example. $\beta_{10} = \beta_{20} = 0$. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 10, p. 32, Feb. 1981, © 1981 IEEE.)

where

$$A_i(z) = \frac{\alpha_{i0} + \alpha_{i1}z^{-1} + \alpha_{i2}z^{-2}}{1 + \alpha_{i3}z^{-1} + \alpha_{i4}z^{-2}}$$

The storage allocation scheme of Figure 13-7 may also be used in this case.

Example 13.4

Consider the eighth-order digital filter designed in Ref. 5 and examined in Ref. 6:

$$D(z) = S_0 \prod_{i=1}^4 A_i(z) \quad (13-26)$$


```

;
; PARALLEL IMPLEMENTATION OF A FOURTH-ORDER FILTER
;
;
; MAIN PROGRAM-CALLS INTEL 8086 SUBROUTINES
FILTER 1:  CALL  INIT          ; INITIALIZE A/D, D/A
                                   ; AND VARIABLES
F1_LOOP:   CALL  INPUT         ; GET SAMPLE FROM A/D
          MOV   BX, AX         ; SAVE X
          IMUL  B0             ; BETA0*X/4
          SAL   DX, 2          ; BETA0*X
          MOV   TEMP, DX       ; SAVE
          MOV   AX, BX         ; RESTORE X(k)
          MOV   CX, #1         ; CALCULATE 1 STAGE
          CALL  OUTP_1D        ;
          SAL   AX, 3          ; BETA1 = -8
          NEG   AX
          ADD   AX, TEMP       ; INCLUDE IN OUTPUT
          MOV   TEMP, AX       ; SAVE
          MOV   AX, BX         ; RESTORE X(K)
          MOV   CX, #1
          CALL  OLP_1D         ; CALCULATE STAGE 2 WITH
                                   ; SI = 2 FROM 1st STAGE
          SAL   AX, 2          ; BETA2 = -4
          NEG   AX
          ADD   AX, TEMP       ; CALCULATE Y
          CALL  OUTPUT         ; SEND Y TO D/A
          MOV   CX, #2
          CALL  DELAY_1D       ; TIME-DELAY
          MOV   CX, #2
          CALL  PRE_1D         ; PRE-PROCESS
          JMP   F1_LOOP        ; CONTINUE
;
;
;
; INIT:   CLEAR M1, M2, T1, T2
;         SET UP A/D, D/A DEVICES
INIT:     MOV   AX, #0
          MOV   CX, #11
          LEA   DI, M0
          REP
          STOW                      ; PUT 0 INTO ALL RAM
          RET
;
;
;

```

Figure 13-9 Parallel implementation of a fourth-order filter.

```

; INPUT-ASSUME MEMORY-MAPPED I/O
; A/D at 60H, STATUS AT E0H
INPUT:  MOV    AX, #0000H    ; START-CONVERSION
        OUT    60H, AL
IN_LP:  IN     AL, 0E0H      ; CHECK END-CONVERSION
        AND    AL, #04H
        JZ     IN_LP        ; WAIT UNTIL READY
        IN     AL, 60H      ; GET SAMPLE FROM A/D
        RET

;
;
;
; OUTPUT to D/A AT 80H
OUTPUT: OUT    80H, AL      ; y(k) TO D/A
        RET

;
;
;
; COEFFICIENT STORAGE
B0:     DW     1            ; BETA_0

A0:     DW     0, 0         ; BETA 10 = BETA 20 = 0
                          ; USED BY OUTP_1D

; HALF VALUES OF A1, A2, B1, B2 USED IN PRE_1D
A1:     DW     20709, -28631, -29868, 14678 ; A1, A2, B1, B2 FOR
                                          ; PRE_1D-STAGE 1
        DW     27416, -25710, -19747, 7270  ; A1, A2, B1, B2 FOR
                                          ; PRE_1D-STAGE 2

; VARIABLES
M0:     DW     2DUP(0)      ; M(K)-BOTH STAGES
M1:     DW     2DUP(0)      ; M(K-1)-BOTH STAGES
M2:     DW     2DUP(0)      ; M(K-2)-BOTH STAGES

; TEMPORARY STORAGE
T1:     DW     2DUP(0)      ; T1-BOTH STAGES
T2:     DW     2DUP(0)      ; T2-BOTH STAGES
TEMP:   DW     0           ; OUTPUT TEMP STORAGE

```

Figure 13-9 (continued)

where

$$S_0 = 0.4383164$$

$$\alpha_{10} = \alpha_{12} = 0.7619852 \quad \alpha_{13} = -0.90191$$

$$\alpha_{11} = -0.2727907 \quad \alpha_{14} = 0.66204$$

$$\alpha_{20} = \alpha_{22} = 0.1963670 \quad \alpha_{23} = -0.59971$$

$$\alpha_{21} = -0.0372331 \quad \alpha_{24} = 0.96329$$

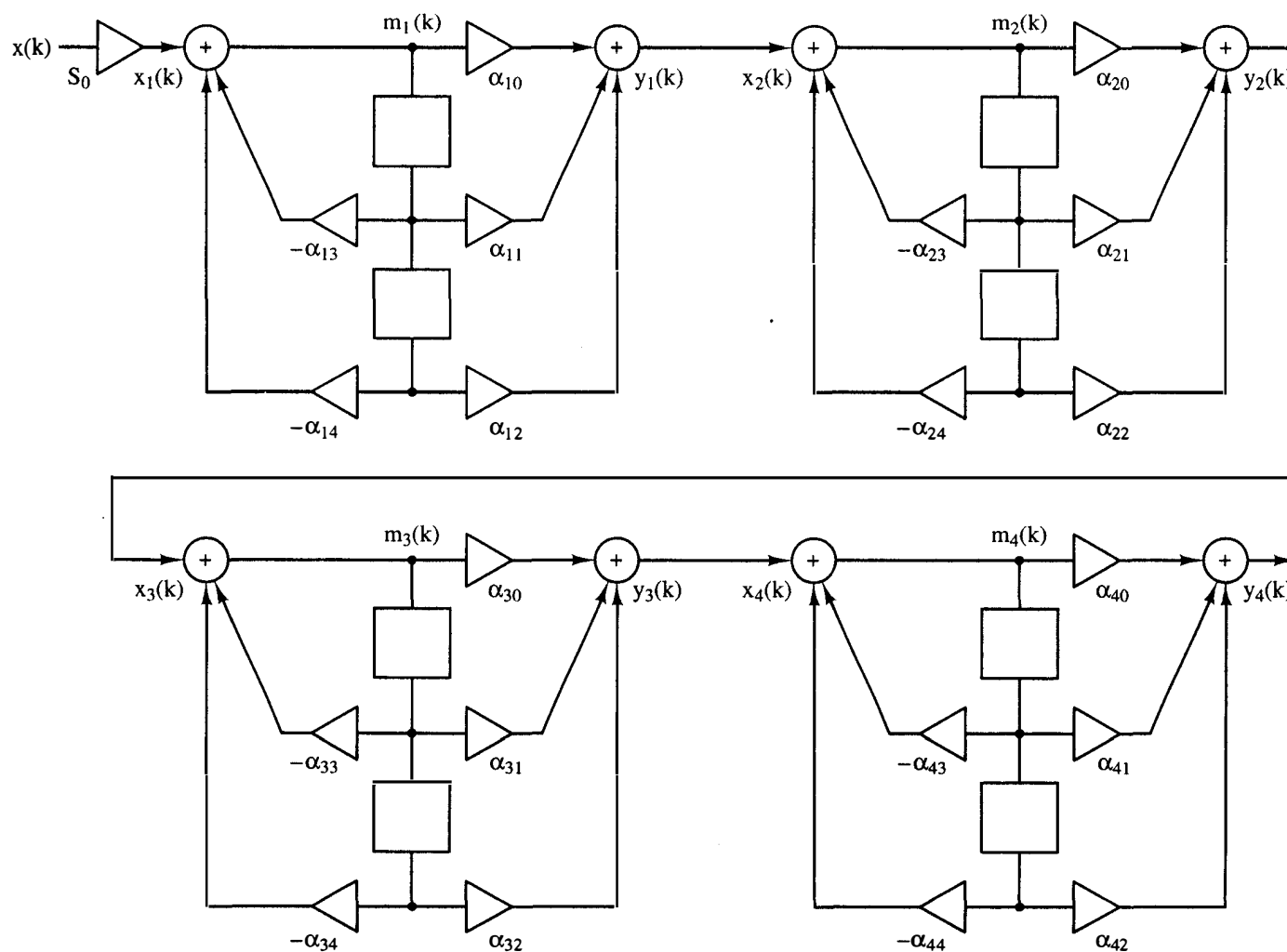


Figure 13-10 Eighth-order example. (From H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Fig. 1, p. 33, Feb. 1981, © 1981 IEEE.)

$$\begin{aligned}
 \alpha_{30} &= \alpha_{32} = 0.2791792 & \alpha_{33} &= -1.17844 \\
 \alpha_{31} &= 0.4518543 & \alpha_{34} &= 0.42357 \\
 \alpha_{40} &= \alpha_{42} = 0.4660647 & \alpha_{43} &= -0.68404 \\
 \alpha_{41} &= 0.1668512 & \alpha_{44} &= 0.85862
 \end{aligned}$$

These coefficients were first designed in Ref. 5. Then Ref. 6 optimized the ordering and pairing to minimize the output round-off noise. Here we have changed the scaling constants (reflected in the numerator terms) to impose a signal limit of 90 percent of full scale at every internally constrained point in the filter for a full-scale step input at $x(k)$. The block diagram of the filter using 1D modules is shown in Figure 13-10.

The Intel 8086 routines to implement Figure 13-10 are shown in Figure 13-11.

```

;
; INTEL 8086 MAIN
FILTER:      CALL    INIT          ; SEE PARALLEL EXAMPLE
FLOOP:      CALL    INPUT        ; GET X(K): SEE PARALLEL EXAMPLE
            IMUL    S0           ; S0*X(K)/4
            SAL     DX, 2        ; S0*X(K)
            MOV     AX, DX
            MOV     CX, #4       ; DO 4 STAGES
            CALL    OUTP_1D      ; COMPUTE Y(K)
            CALL    OUTPUT      ; OUTPUT Y(K): SEE PARALLEL EXAMPLE
            MOV     CX, #4
            CALL    DELAY_1D
            MOV     CX, #4       ; DO 4 STAGES
            CALL    PRE_1D       ; PREPROCESS FOR NEXT SAMPLE
            JMP     FLOOP        ; NEXT SAMPLE
;
;
;
; 1D FILTER COEFFICIENT STORAGE FOR 4 STAGES
S0:          DW      7181                ; S0/2
A0:          DW      12484, 3217, 4574, 7636 ; ALPHA 10, 20, 30, 40
A1:          DW      -4469, 12484, -14777, 10847 ; ALPHA 11, 12, 13, 14
            DW      -610, 3217, -9826, 15783 ; ALPHA 21, 22, 23, 24
            DW      7403, 4574, -19308, 6940 ; ALPHA 31, 32, 33, 34
            DW      2734, 7636, -11207, 14068 ; ALPHA 41, 42, 43, 44
; VARIABLE STORAGE
M0:          DW      4DUP(0)             ; M(k)
M1:          DW      4DUP(0)             ; M(k-1)
M2:          DW      4DUP(0)             ; M(k-2)
; TEMPORARY STORAGE
T1:          DW      4DUP(0)             ; TEMP STORAGE
T2:          DW      4DUP(0)             ; TEMP STORAGE

```

Figure 13-11 Cascade implementation of an eighth-order filter.

13.6 COMPARISON OF STRUCTURES

Here we use the example of Section 13.5 to compare structures. Suppose that the cascade of four second-order modules were composed of 2D, 3D, 4D, 1X, or 2X structures. How do their sampling rates compare? Table 13-1 shows that for an eighth-order filter, the 3D structure is fastest, while the 2X is slowest. The four direct structures are all approximately equal in sampling rate, while the cross-coupled structures are both considerably slower, requiring more multiply operations.

13.7 LabVIEW [7,8]

LabVIEW is a program development environment that uses a graphical programming language rather than the traditional textual languages (C, assembly, FORTRAN, Pascal, etc.). LabVIEW relies on icons and graphical symbols to specify programming actions. The programs resemble block diagrams or data flow graphs. LabVIEW programs are called virtual instruments (VIs) because they appear to simulate physical instruments by having a front panel and a block (or wiring) diagram. Extensive libraries of functions and subroutines are supplied with the software package. In this section we use LabVIEW to design a program for the digital filter of (13-24) and Figure 13-8.

Virtual Instruments

Figure 13-12 shows a typical example of a program in LabVIEW [9]. This VI simulates the transmission and reception of a pulse sent across a noisy channel. The *Front Panel* is used to describe all the input and output parameters and variables for the program. Its accompanying *Block Diagram* shows the data flow and operators of the program. The Block Diagram is constructed using hardware-language schematic capture techniques. A wiring tool is used to connect (or wire) elements in the

TABLE 13-1 COMPARISON OF EIGHTH-ORDER CASCADED FILTERS USING 1D, 2D, 3D, 4D, 1X, AND 2X STRUCTURES (FOR INTEL 8086 ROUTINES) (μ s)

Routine	Structure YY					
	1D	2D	3D	4D	1X	2X
OUTP_YY	176.0	161.6	163.6	176.0	161.6	161.6
POST_YY	0	623.2	0	582.4	848.0	595.2
DELAY_YY	32.8	0	27.4	18.4	32.4	0
PRE_YY	582.4	0	565.6	0	0	296.8
TOTAL	791.2	784.8	756.6	776.8	1042.0	1053.6

Source: H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, Table 5, p. 31, Feb. 1981, © 1981 IEEE.

Connector Panel

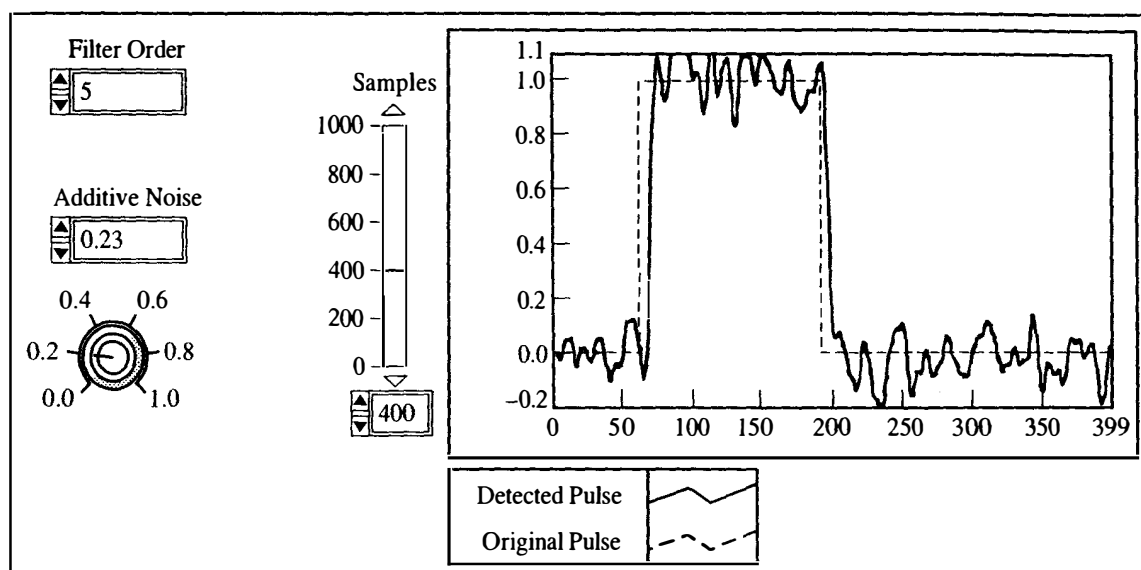


Pulse Demo.vi

The goal of this example is to demonstrate the power of LabVIEW as a tool for simulation.

This example simply simulates a transmission and receiver system.

Front Panel



Block Diagram

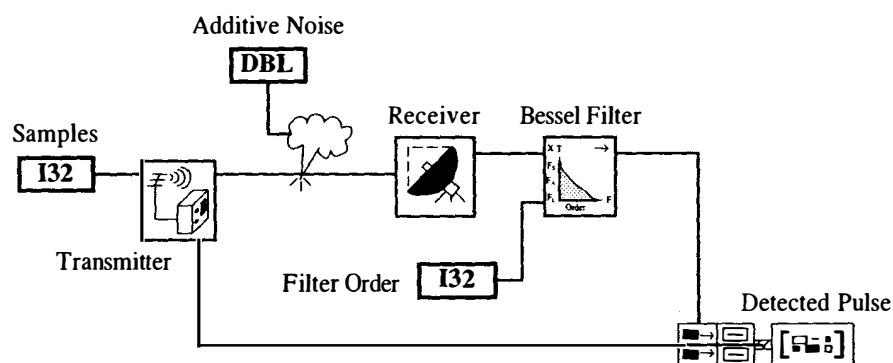


Figure 13-12 Pulse transmit/receive example.

Block Diagram. In this example, the number of data samples is an input parameter specified on the Front Panel. This parameter is passed to a sub-VI (a subroutine called Transmitter) that generates a pulse with the specified number of samples and modulates the pulse for transmission to the sub-VI named Receiver. However, another sub-VI shaped like a cloud intercepts the modulated pulse and adds white noise to simulate atmospheric interference. This noise sub-VI accepts a parameter

(Additive Noise) from the Front Panel that specifies the absolute value of the amplitude of the additive noise source. Thus, from the Front Panel one can try different levels of noise to see the performance of the system.

The noisy pulse is next transmitted to the sub-VI named Receiver. This subroutine demodulates the noisy signal and sends it to the sub-VI named Bessel Filter. The Bessel Filter reduces the noise in the signal and passes it on to the Waveform Graph on the Front Panel for observation. Note that the Filter Order is another control parameter from the Front Panel. Various combinations of Filter Order and Additive Noise amplitude can be attempted by the user of the program.

The Waveform Graph is labeled Detected Pulse on the Block Diagram. The block preceding the Graph builds a two-dimensional array of the original and detected pulses so that they can be plotted on the same graph. The legend on the Front Panel shows that the original pulse is a dotted line while the detected pulse is solid. Each of the sub-VIs in this example is another program that itself has sub-VIs. Thus hierarchy can be used to simplify programs and aid in documentation.

Arithmetic Operations

In Figure 13-12 we illustrated how to build a VI using sub-VIs. The sub-VIs are constructed using a number of functions provided in a series of LabVIEW function menus. Figure 13-13 illustrates a six typical functions from the Arithmetic Menu. The top two operators are used to add and multiply two scalar numbers, vectors, or arrays. The third takes the absolute value of a single input, which may also be a scalar number, vector, or array. The fourth and fifth perform the logical AND and logical exclusive-OR operations on two numbers or Boolean variables. The last example illustrates a random number generator. The function has no inputs and generates a double-precision floating-point number between 0 and 1.

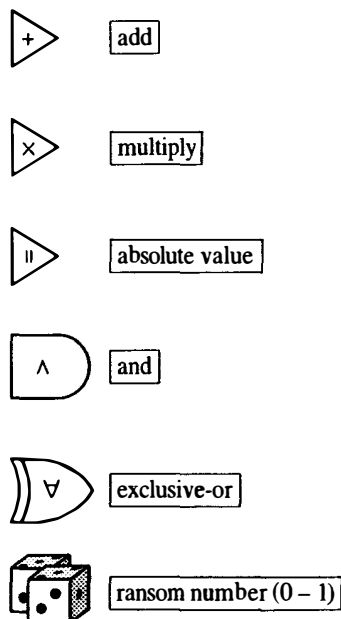


Figure 13-13 Example functions: Arithmetic Menu.

Programming Structures and Constants

Computer programs not only need arithmetic and logical operations, but also must have control structures to guide program execution. In LabVIEW four program control structures are available under the Structs & Constants Menu (see Figure 13-14). The While Loop structure has two symbols located in its interior. The symbol i represents the iteration terminal. This terminal indicates the number of iterations of the loop and begins with the value 0. The other symbol is called the conditional terminal. The While Loop continues to execute any sub-VIs located inside its border until a Boolean value wired to the conditional terminal is FALSE.

The For Loop also has two symbols inside its border. The symbol N is the count terminal. An externally supplied integer connected to the count terminal can be used to specify the number of times any sub-VIs located inside the loop's border will be executed. The symbol i represents the iteration terminal and assumes the values 0 to $N-1$ as the loop executes its iterations.

The Sequence structure resembles a frame of photographic film and consists of one or more subdiagrams, or frames, that execute sequentially. Each frame has a frame number in its upper border. This structure controls the sequence of program execution (frame 0, frame 1, etc.) and provides mechanisms to pass variables from frame to frame.

The fourth flow control mechanism shown in Figure 13-14 is the Case structure. It can have two or more subdiagrams, but only one executes, depending on an

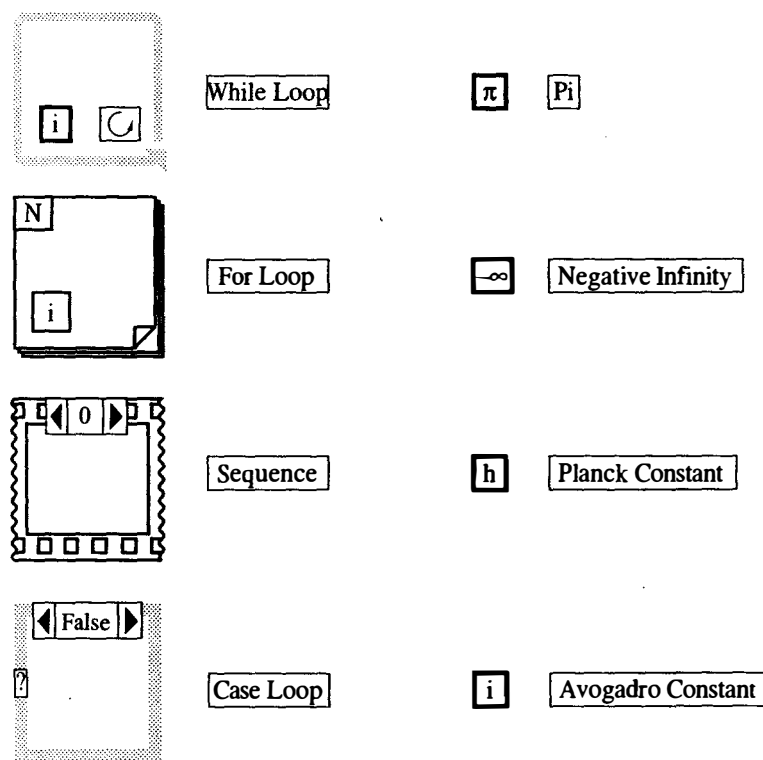


Figure 13-14 Example functions: Structs & Constants Menu.

external value connected to the selection terminal (labeled with a ?) located in its left border. The example in Figure 13-14 shows a Boolean Case structure that has two subdiagrams. A Boolean value connected to the selection terminal will have two values (TRUE or FALSE) controlling which of the two subdiagrams will execute.

Finally, Figure 13-14 illustrates a few of the many numerical constants available from the Structs & Constants Menu. Each of these examples may be used as a source for these commonly used numbers.

Other Available Functions

LabVIEW has a generous supply of advanced functions and sub-VIs for performing trigonometric and logarithmic operations, comparisons, data structure conversions, string and array manipulation, file I/O, signal analysis, data acquisition, GPIB handlers, network routines, and error-handling utilities. The interested reader should contact National Instruments for a demonstration kit.

Data Acquisition Hardware

LabVIEW is designed to operate in conjunction with a wide range of data acquisition hardware from many different manufacturers. National Instruments also offers a complete line of products for Apple Macintosh and IBM PC-compatible computers. The lab-NB series is an inexpensive way to start on the Macintosh; the lab-PC+, on IBM PC compatibles. Drivers for both of these boards are provided in the student version of LabVIEW [10].

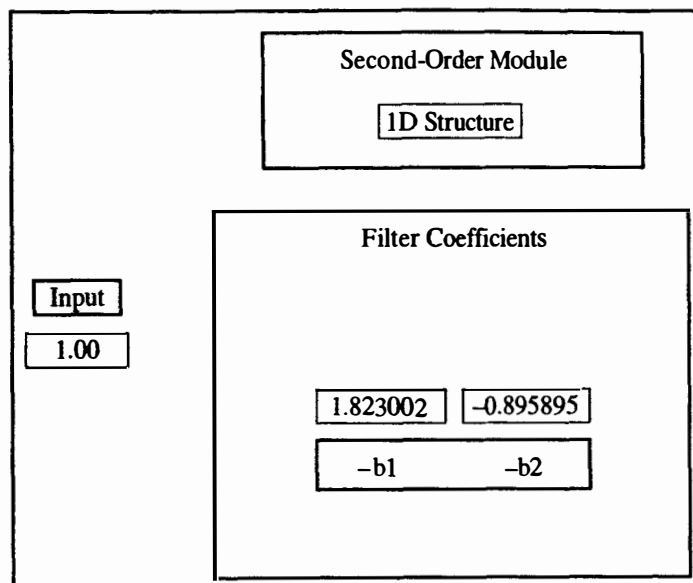
Second-Order Module

Let us use LabVIEW to build a program to implement the second-order module for the 1D structure shown in Figure 12-2a and (12-7). From Figure 12-2a we note that five multiply operations and two summing junctions must be implemented. To accomplish this, we may use the Add and Multiply functions illustrated in Figure 13-13. But we also need to use a two-element shift register to perform time delay. Shift registers in LabVIEW are implemented as special features of the While and For Loops. Special symbols are added to the border of the Loop as shown in Figure 13-15. This figure shows an implementation of the first equation in (12-7):

$$m(k) = x(k) - b_1 m(k-1) - b_2 m(k-2)$$

The Front Panel has three control inputs: the Input signal representing $x(k)$ and the two filter coefficients, $-b_1$ and $-b_2$. The Block Diagram uses two multiply and two add operations to calculate $m(k)$. The value of $m(k)$ is supplied to the upward-pointing triangle in the right border of the For Loop. Upon the next iteration of the loop, this value is delayed two time periods through the downward-pointing triangles in the left border. The For Loop executes one time as instructed by the constant 1 connected to N.

Front Panel



Block Diagram

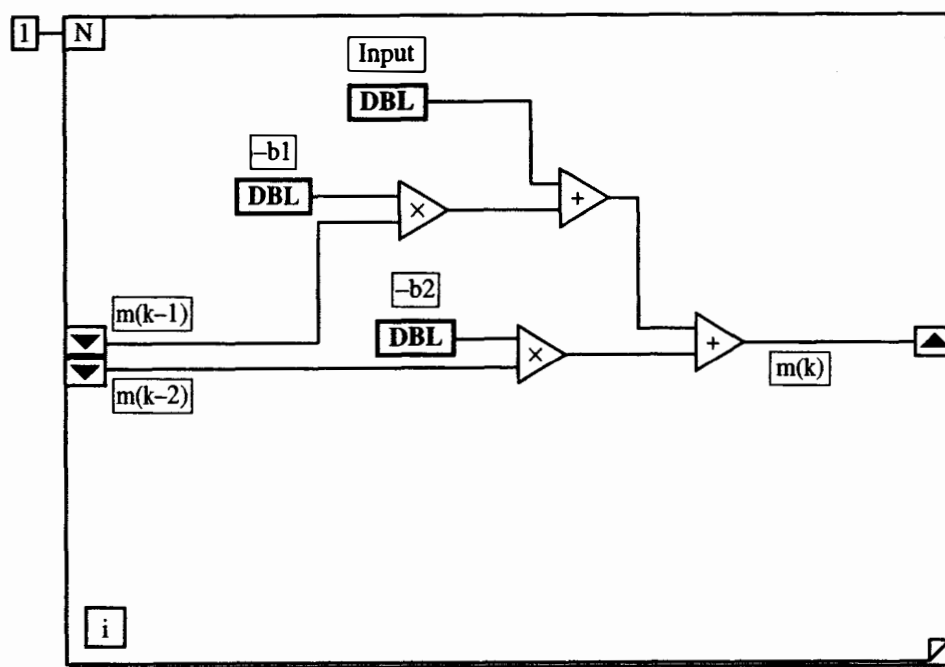
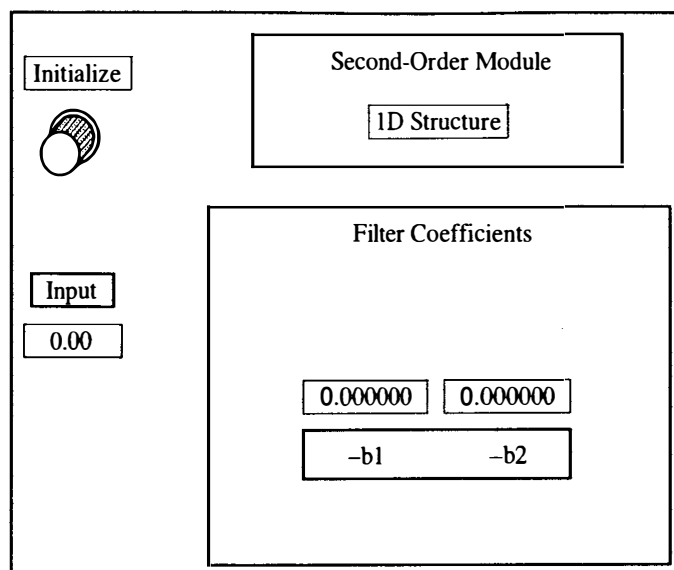


Figure 13-15 Shift registers.

How can we initialize the shift register? Figure 13-16 illustrates one technique. An Initialize control button is placed on the front panel. It supplies a Boolean value TRUE when pressed. On the Block Diagram, Initialize is connected to a multiplexer that connect the upper channel (the one with a value of zero) to its output when the Boolean value is TRUE. When the value is FALSE the signal $m(k)$ is supplied to

Front Panel



Block Diagram

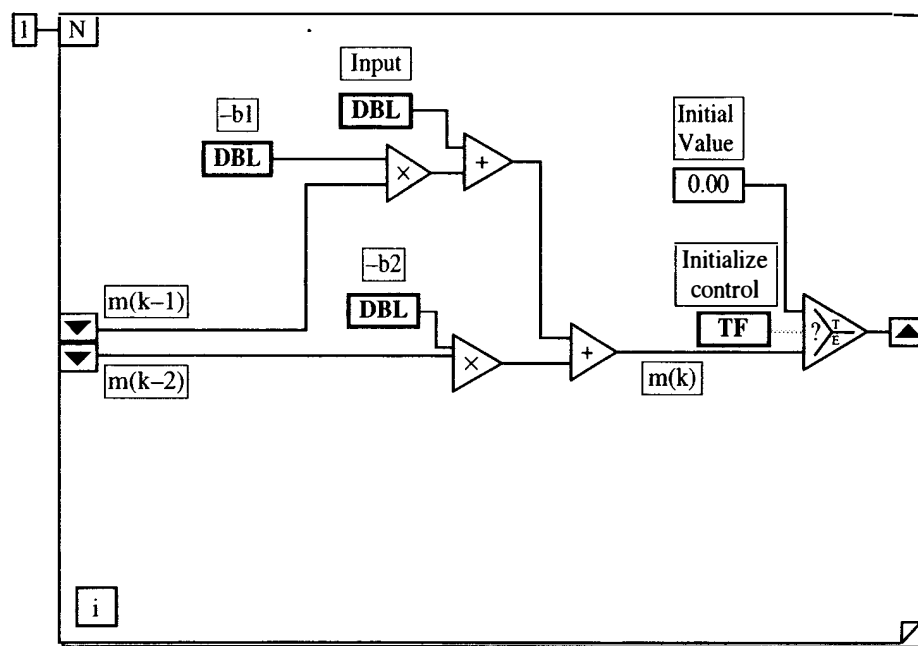


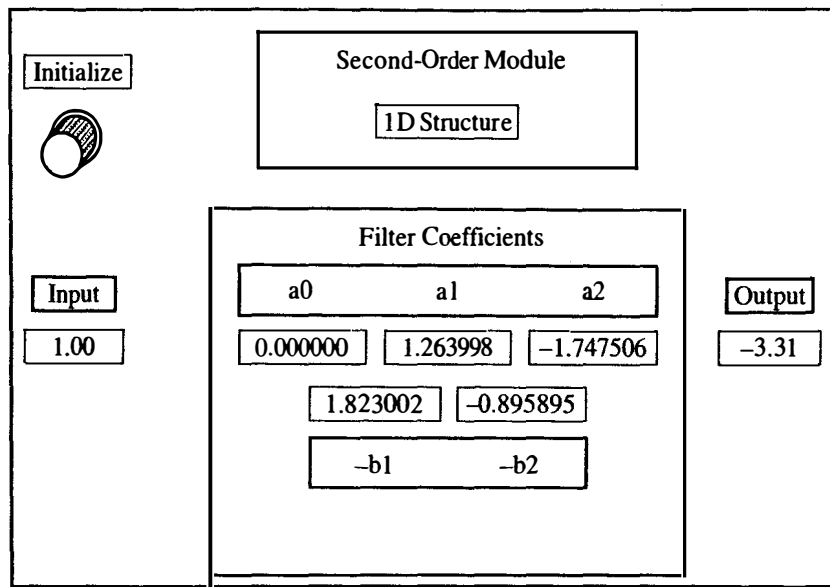
Figure 13-16 Initialization.

the shift-register input. So pressing the Initialize control button forces zeros into the shift register.

Now we can finish the programming of the second-order module by adding multiply and add operators to implement the second equation in (12-7):

$$y(k) = a_0 m(k) + a_1 m(k-1) + a_2 m(k-2)$$

Front Panel



Block Diagram

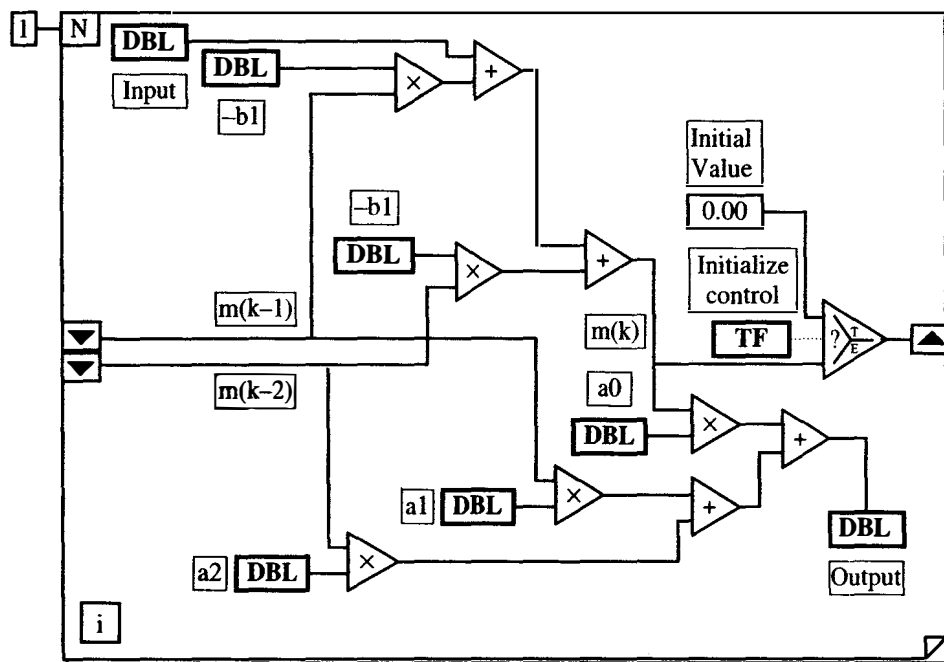


Figure 13-17 Second-order 1D module.

Figure 13-17 shows the completed program. Note that an indicator labeled Output has been incorporated into the front panel. This Output signal represents $y(k)$ in (12-7).

The final step in programming the module is to give it a unique icon so that it can be used as a sub-VI in other programs. Figure 13-18 displays the Icon and

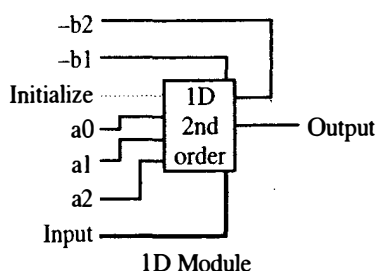


Figure 13-18 Module icon and its connections.

Connector Pane designed for this program. The Connector Pane illustrates the wiring sites for the Icon. Note that the program accepts seven input values and generates one output value.

Fourth-Order Example

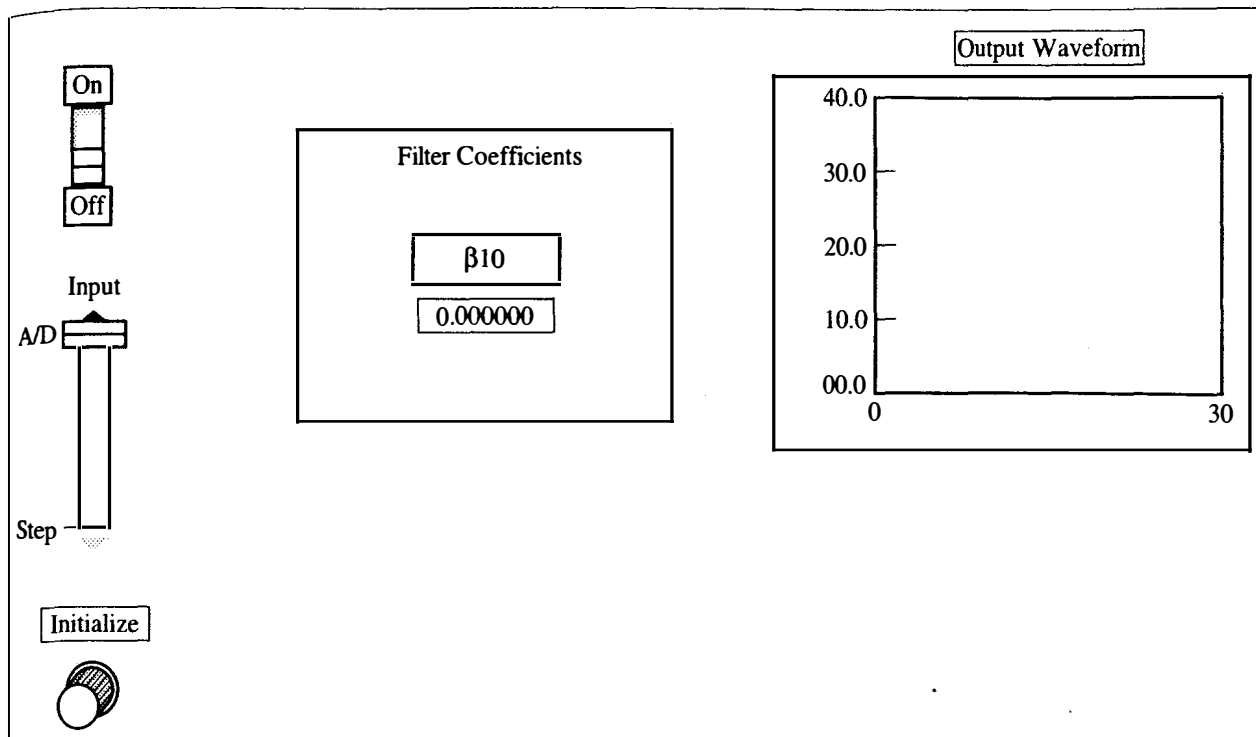
As a final example of LabVIEW programming, let us now use the second-order module to implement the fourth-order digital filter of Figure 13-8 and (13-24). Consider the partial solution shown in Figure 13-19. First note that a While Loop is used as the overall controlling structure with the On/Off switch from the Front Panel connected to its conditional terminal. While the switch is in the On position, the program executes. Placing the switch in the Off position terminates the While Loop.

Another interesting feature of this partial program is the symbol located along the lower Loop border labeled Millisecond Timer. This function forces the Loop to synchronize its iterations with an internal millisecond clock. In this case the waiting period is 100 ms, which sets the sampling rate to 10 Hz.

Next examine the sub-VI "1D 2nd order" in the center of Figure 13-19. This is the icon for the second-order module sub-VI of Figure 13-17. Only three of its weight required signals have been wired. From the Connector Pane of Figure 13-18 we observe that the three wired connections are to the Initialize, a0, and Input control terminals.

The input to the filter is either a unit step or the A/D converter as indicated by the slide control switch on the Front Panel. When the slide switch is in the lower position labeled Step, it supplies a zero value to the $=0$ comparison operator in the Block Diagram. In this case a TRUE value is furnished to the multiplexer connecting its upper input to its output. The constant 1 is consequently supplied representing a unit step input for the filter. When the slide switch on the Front Panel is in the upper position labeled A/D, the sub-VI "AI ONE PT" (analog single point) furnishes a single sample from the A/D converter for the filter input variable. In addition to its previous role of initializing the internal shift registers of the second-order filter module, notice that the Initialize control button on the front panel has been wired to force a zero into the digital filter's input. This forces all the filter's internal signals to zero in just two iterations of the While Loop.

Front Panel



Block diagram

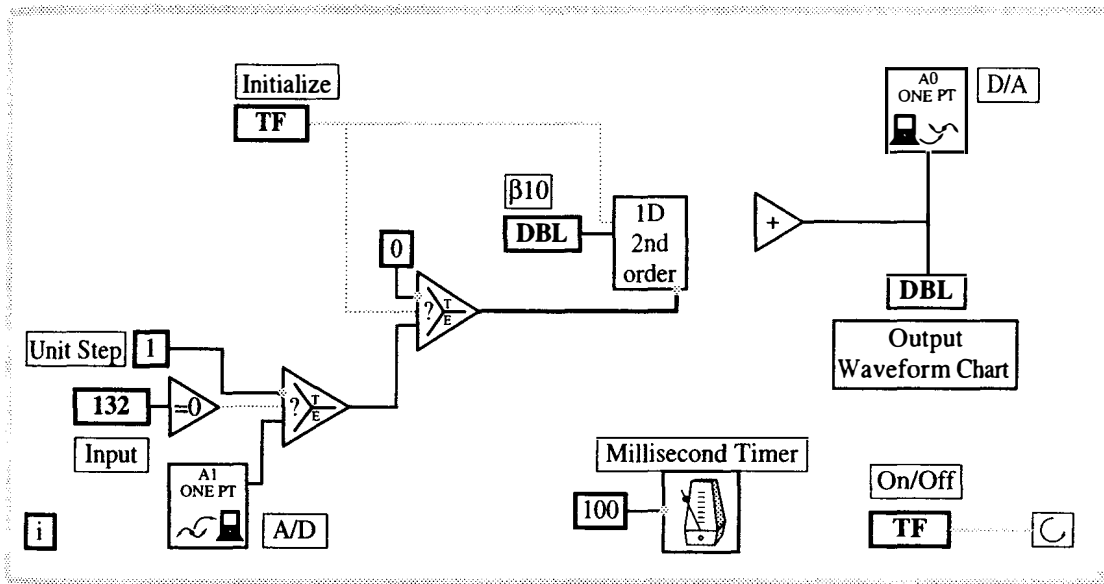
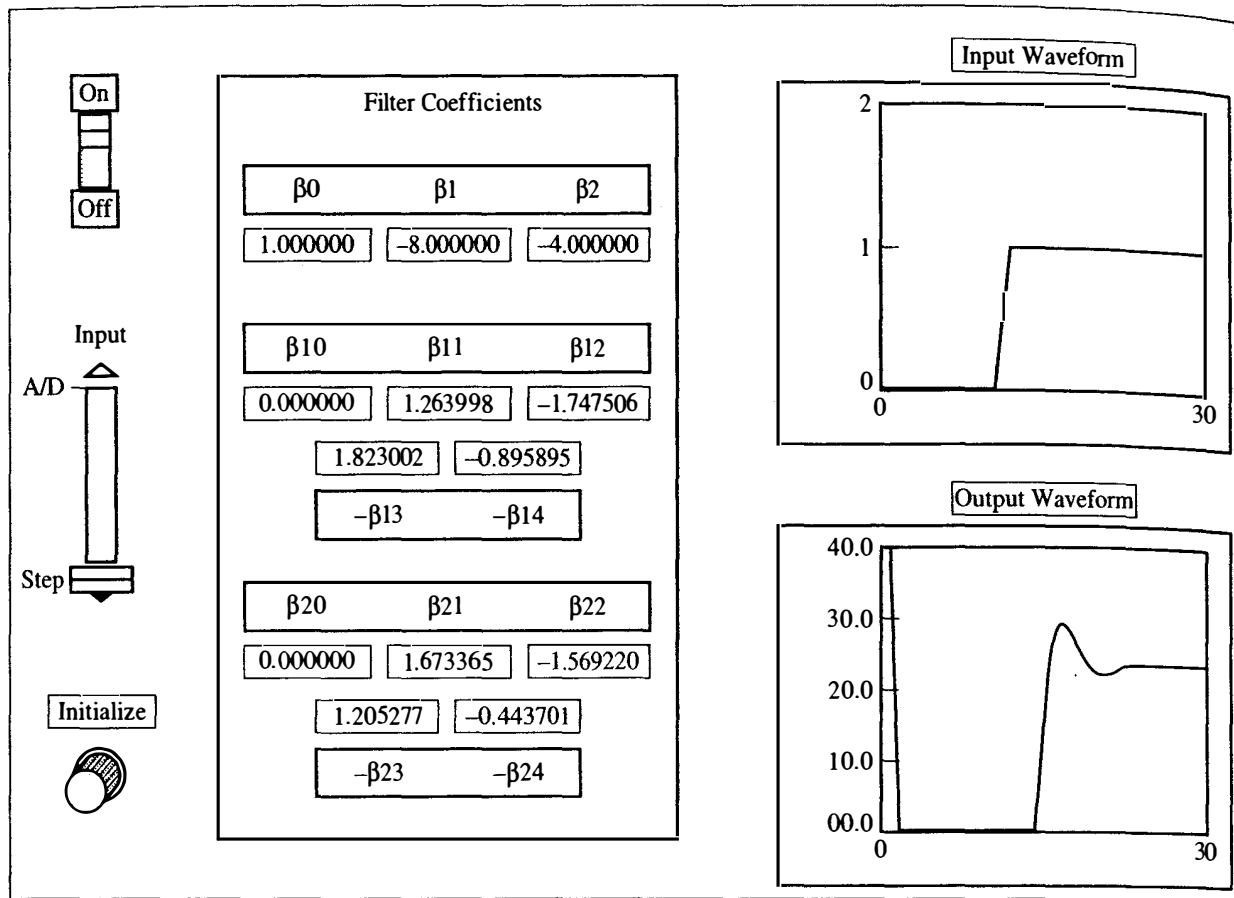


Figure 13-19 Partial solution.

The output of the fourth-order filter is produced by an Add operator and is sent to the D/A converter (sub-VI "AO ONE PT") and a Waveform Chart on the Front Panel. The Waveform Chart for the signal Output allows the operator to visualize the filter's performance in real time.

Now we are ready to complete the program (see Figure 13-20). All 13 of the

Block diagram



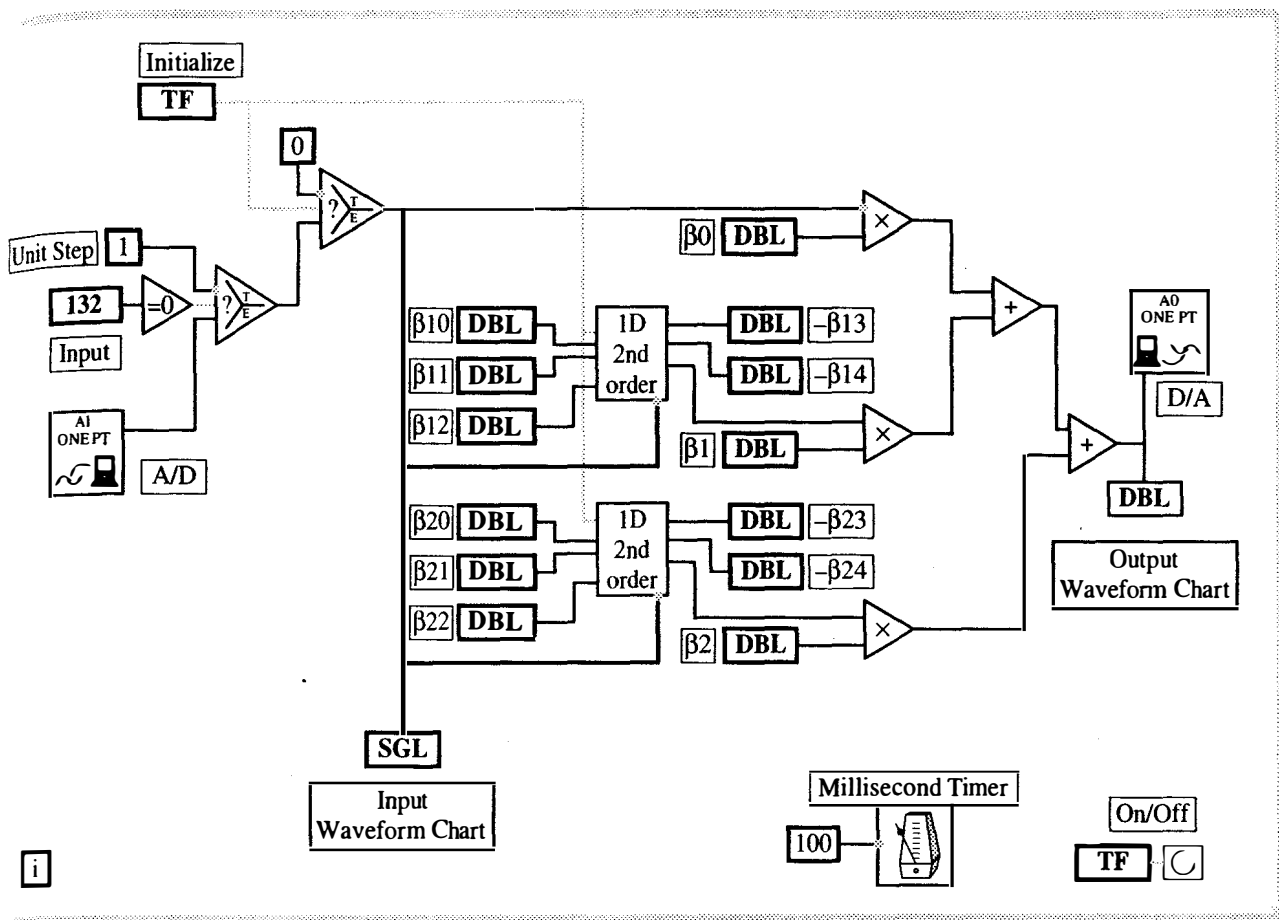
(a)

Figure 13-20 Fourth-order digital filter.

filter's coefficients have been added to the Front Panel as well as a Waveform Chart for the filter's input signal. On the Block Diagram notice that the second "1D 2nd order" sub-VI has been placed and both second-order modules are wired in the configuration of Figure 13-8. The data displayed on the Waveform Charts on the Front Panel show a typical execution of the program. These traces were obtained using the following steps. First the Input slide switch was placed in the Step position. Then the Initialize button was pushed. Next the On/Off switch was placed in the On position. Notice that the initialization of the filter occurred in two iterations as expected. After about 1 s, the Initialize button was released allowing the unit step to be applied to the filter's input. Note that the step response has reached steady state in about 15 iterations (1.5 s).

This completes our introduction to LabVIEW. LabVIEW is a very powerful graphical programming language that is finding wide application in digital control systems. Large numbers of VIs are available. Most GPIB instrument manufacturers

Block diagram



(b)

Figure 13-20 (continued)

are now providing LabVIEW sub-VIs for control of their instrumentation products. Its use in teaching laboratories is also expanding rapidly.

13.8 SUMMARY

In this chapter we have investigated the computer implementation of digital filters. Emphasis has been placed on assembly language implementations for the INTEL 80×86 family of processors that are widely available in IBM PC-compatible machines. We have also illustrated the implementation of digital filters using the graphical programming language LabVIEW. LabVIEW has great flexibility but executes more slowly than the assembly language routines for the 80×86 . So for fast sampling rates (greater than 1 kHz), designers must run benchmark tests on the digital control system's computer platform to verify that LabVIEW will meet system specifications.

REFERENCES

1. B. B. Brey, *The Intel Microprocessors*, 2d ed. New York: McMillan Publishing Co., Inc., 1991.
2. D. Alpert and D. Avnon, "Architecture of the Pentium Microprocessor," *IEEE MICRO*, pp. 11–21, June 1993.
3. H. T. Nagle, Jr., and V. P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers," *IEEE MICRO*, Vol. 1, No. 1, pp. 23–41, Feb. 1981.
4. H. T. Nagle, Jr., and C. C. Carroll, "Realization of Digital Controllers," *Proc. IFAC Symp. Autom. Control Space*, Armenia, USSR, Aug. 1974.
5. B. Gold and C. M. Radar, *Digital Processing of Signals*. New York: McGraw-Hill Book Company, 1969.
6. S. Y. Huang, "On Optimization of Cascade Fixed-Point Digital Filters," *IEEE Trans. Circuits Syst. (Letters)*, Vol. CAS-21, pp. 163–166, Jan. 1974.
7. J. M. Jagadeesh and Y. Wang, "LabVIEW Product Review," *IEEE Computer*, pp. 100–103, Feb. 1993.
8. *LabVIEW User Manual*, National Instruments Part Number 320591-01, Aug. 1993.
9. *Analysis Examples, LabVIEW 3.0 Distribution Software*, National Instruments Part Number 776762-01, Oct. 15, 1993.
10. *Student Version of LabVIEW*, National Instruments, Austin, TX, 1994.

PROBLEMS

- 13-1. Given

$$D_1(z) = \frac{1 + z^{-1}}{1 - 0.97337z^{-1}}$$

Find the stored value of these coefficients for the Intel 8086 1D Structure implementation.

- 13-2. Repeat Problem 13-1 for

$$D_2(z) = \frac{1 - 1.9661z^{-1} + z^{-2}}{1 - 1.9654z^{-1} + 0.99930z^{-2}}$$

- 13-3. Repeat Problem 13-1 for

$$D_3(z) = \frac{1 - 1.9661z^{-1} + z^2}{(1 - 0.22433z^{-1})(1 - 0.52749z^{-1})}$$

- 13-4. Compute the frequency response of

$$D(z) = D_1(z)D_2(z)D_3(z)$$

Compare the result with Figure 11-15.

- 13-5. Illustrate how the direct structure routines for the Intel 8086 calculate

$$aX = (0.435102)(0.713001)$$

where a is a filter coefficient and X is a signal variable.

- 13-6. Generate a LabVIEW program to implement the digital filter of (13-26) using second-order 1D modules.

Finite-Wordlength Effects

14.1 INTRODUCTION

In Chapters 4 through 12 we have presented the analysis and design of discrete-time linear systems. Signal variables and system coefficients were *real* numbers; that is, they were continuous (analog) variables and fixed constants without restriction on their specific values. However, in Chapter 13 we implemented in digital hardware the digital filters designed earlier. In these practical implementations, the values of signal variables and filter coefficients are restricted to a finite set of discrete magnitude values. In Chapter 13 we used a fixed-point number system (two's complement). Other researchers have described distributed arithmetic, signed-logarithm, canonical signed-digit code, input-scaled floating point, residue number systems, Fibonacci numbers, and Fermat transforms to implement digital filters. Floating point [1] has also been used in implementing digital systems. Fixed-point number systems are the most economical and generally applicable approach toward implementing digital filters. In this chapter we examine fixed-point number systems and analyze their effectiveness in implementing digital filters. Specifically, we will analyze coefficient quantization, filter input quantization, product quantization, round-off noise, limit cycles due to product quantization, overflow properties, signal dynamic range, and signal-to-noise ratios.

14.2 FIXED-POINT NUMBER SYSTEMS

The choice of a number system to implement a digital filter greatly affects the filter's performance. The accuracy with which coefficients and signal variables may be

represented is directly related to the quantization properties, overflow characteristics, and dynamic range of the number system. In this section we describe two different number systems that have been used in implementing digital filters.

In what follows, we assume that a real number x is to be represented as a finite number of bits in a quantized version of x , say $Q(x)$. The accuracy of the representation may be measured by the error

$$e \triangleq Q(x) - x \quad (14-1)$$

To simplify the notation we will assume that each number x [and hence $Q(x)$] lie in the range

$$0 \leq |x| \leq 1 \quad (14-2)$$

In practice, if numbers are larger than 1, we may normalize them to this range by simply shifting the binary point by some L bits. So, if

$$0 \leq |x'| \leq C \quad (14-3)$$

then

$$0 \leq |Q(x')| \leq C$$

where $C > 1$, there is an L such that

$$Q(x) = 2^L Q(x') \quad (14-4)$$

and

$$0 \leq |Q(x)| \leq 1$$

Unless otherwise stated, we will always assume that numbers in each number system lie in this range.

Signed-Magnitude Number System

The signed-magnitude number system may be used to represent digital filter coefficients and signal variables. In general, any number in the signed-magnitude system may be expressed

$$Q^b(x) = (s.m_1 \ m_2 \ m_3 \ \cdots \ m_b)_{2\text{smns}}$$

where $Q^b(x)$ is a quantized version of a number x

s is the sign bit

$s = 0$ for x positive

$s = 1$ for x negative

m_i are the magnitude bits

$$(.m_1 \ m_2 \ \cdots \ m_b)_2 = |Q(x)|$$

That is, the magnitude of x is approximated by a binary fraction. Here all numbers are normalized such that

$$0 \leq |Q^b(x)| < 1$$

We may also use a series notation for $Q^b(x)$ as follows:

$$Q^b(x) = (1 - 2s) \sum_{i=1}^b m_i \cdot 2^{-i} \quad (14-5)$$

Now what remains is the matter of determining $m_i, i = 1, \dots, b$ given x .

In this section we examine three quantizing methods that have been proposed for use in digital filters: truncation, round-off, and least significant bit 1 (LSB-1).

First let us consider the case of the *truncation* quantizer. In this case the $|x|$ is a positive real number and is converted to a binary fraction as an infinite series, and then the series is truncated at b bits, as shown below:

$$|x| = (\underbrace{.m_1 \ m_2 \ \cdots \ m_b}_{|Q_t^b(x)|} \ m_{b+1} \ \cdots)_2 \quad (14-6)$$

and

$$|Q_t^b(x)| = (.m_1 \ m_2 \ \cdots \ m_b)_2$$

The subscript t indicates truncation and the superscript b indicates the number of bits. Now let us examine the error e_t introduced by the truncation. For $x \geq 0$, $|x| \geq |Q_t^b(x)|$, and

$$\begin{aligned} e_t &= Q_t^b(x) - x \\ &= |Q_t^b(x)| - |x| \\ &= -(.000 \ \cdots \ 0 \ m_{b+1} \ m_{b+2} \ \cdots)_2 \\ &= -2^{-b}(.m_{b+1} \ m_{b+2} \ \cdots)_2 \end{aligned}$$

But $(.m_{b+1} \ m_{b+2} \ \cdots)_2$ is a real number bounded by

$$0 \leq (.m_{b+1} \ m_{b+2} \ \cdots)_2 < 1 \quad (14-7)$$

Therefore,

$$0 \geq e_t > -2^{-b}, \quad x \geq 0 \quad (14-8)$$

For the case in which $x < 0$,

$$\begin{aligned} e_t &= Q_t^b(x) - x \\ &= -(|Q_t^b(x)| - |x|) \end{aligned}$$

so that

$$0 \leq e_t < 2^{-b}, \quad x < 0 \quad (14-9)$$

The quantization characteristic for $Q_t^b(x)$ is illustrated in Figure 14-1a. This nonlinear transfer characteristic introduces many problems in digital filters.

Note that the sign of x determines the value of the truncation error. If we assume that x is a real number and is equally likely to have positive or negative values,

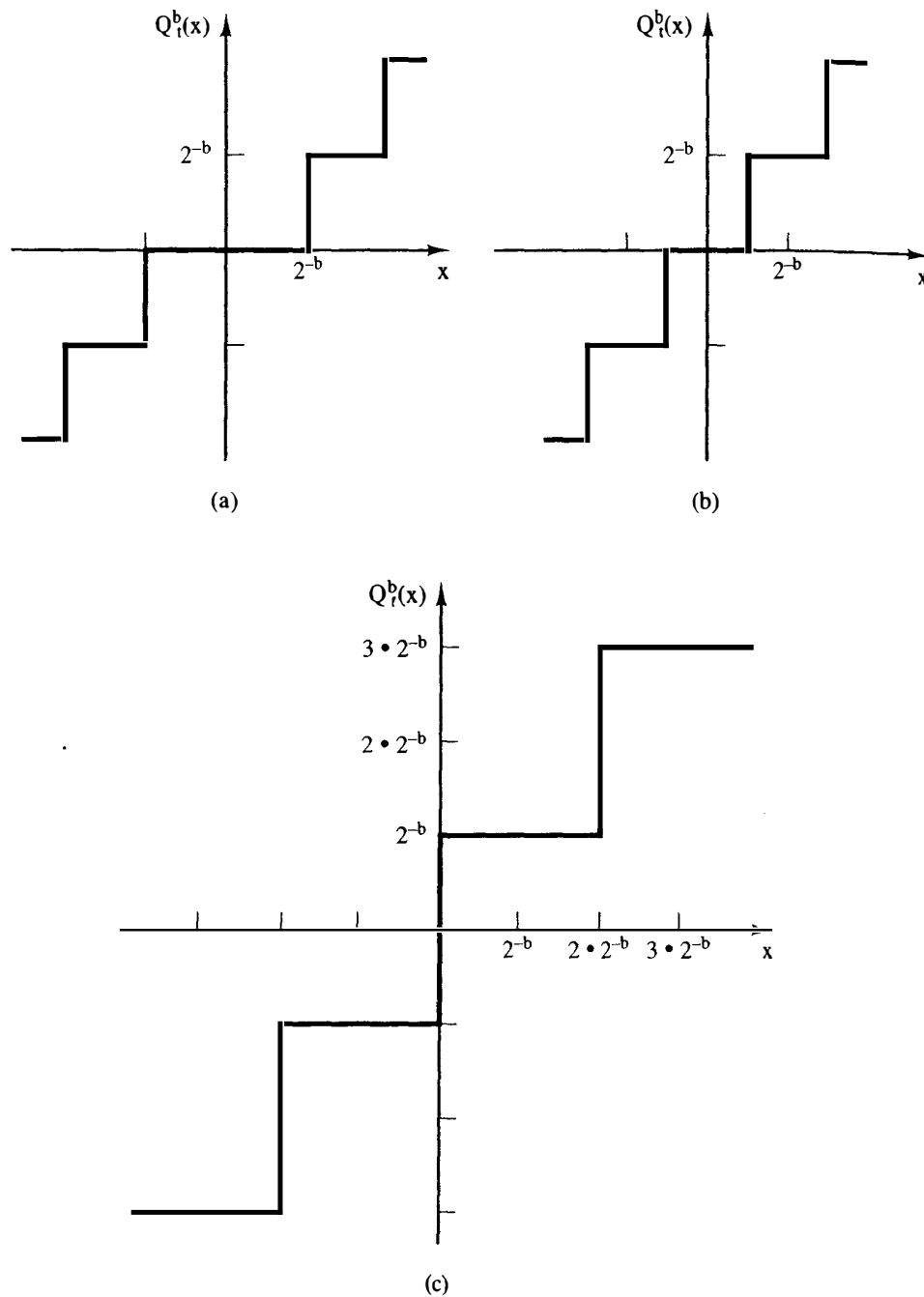


Figure 14-1 Quantizer characteristics for the signed-magnitude number system: (a) truncation; (b) round-off; (c) LSB-1.

the probability density function for e_i is continuous and may be depicted as shown in Figure 14-2a and the noise variance may be approximated by

$$\sigma_{e_i}^2 = \{E[e_i^2] - E^2[e_i]\}_{x \geq 0} + \{E[e_i^2] - E^2[e_i]\}_{x < 0} \quad (14-10)$$

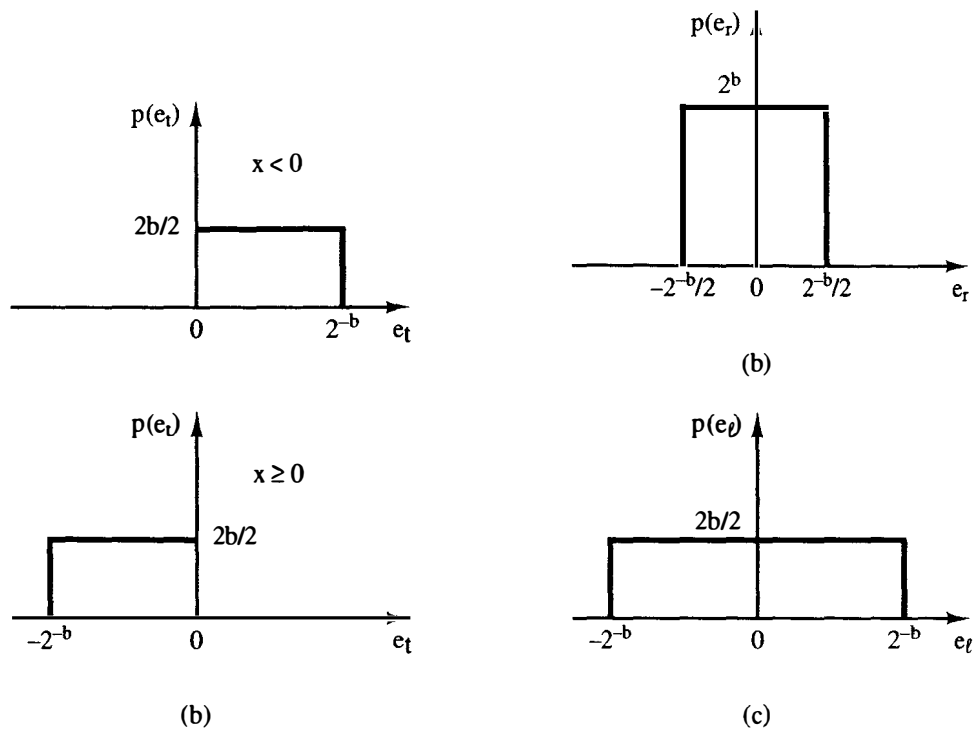


Figure 14-2 Quantification error probability density function for the signed-magnitude number system: (a) truncation; (b) round-off; (c) LSB-1.

But

$$\begin{aligned}
 E[e_t^2]_{x \geq 0} &= \int_{-2^{-b}}^0 e_t^2 \left(\frac{2^b}{2} \right) de_t \\
 &= \frac{2^b}{2} \frac{e_t^3}{3} \Big|_{-2^{-b}}^0 = \frac{2^b}{2} \frac{2^{-3b}}{3} = \frac{2^{-2b}}{6} \\
 E[e_t]_{x \geq 0} &= \int_{-2^{-b}}^0 e_t \left(\frac{2^b}{2} \right) de_t \\
 &= \frac{2^b}{2} \frac{e_t^2}{2} \Big|_{-2^{-b}}^0 = -\frac{2^b}{2} \frac{2^{-2b}}{2} = -\frac{2^{-b}}{4} \\
 E[e_t^2]_{x < 0} &= \int_0^{2^{-b}} e_t^2 \left(\frac{2^b}{2} \right) de_t \\
 &= \frac{2^b}{2} \frac{e_t^3}{3} \Big|_0^{2^{-b}} = \frac{2^{-2b}}{6} \\
 E[e_t]_{x < 0} &= \int_0^{2^{-b}} e_t \left(\frac{2^b}{2} \right) de_t \\
 &= \frac{2^b}{2} \frac{e_t^2}{2} \Big|_0^{2^{-b}} = \frac{2^{-b}}{4}
 \end{aligned}$$

Therefore,

$$\sigma_{e_r}^2 = \left\{ \frac{2^{-2b}}{6} - \left(\frac{2^{-b}}{4} \right)^2 \right\}_{x \geq 0} + \left\{ \frac{2^{-2b}}{6} - \left(\frac{2^{-b}}{4} \right)^2 \right\}_{x < 0} = \frac{5 \cdot 2^{-2b}}{24} \quad (14-11)$$

Next, let us consider the *round-off* quantizer. This quantizer takes a value $|x|$, which may be written as an infinite series in powers of 2, and rounds its value to the nearest term 2^{-b} as follows:

$$|x| = (.n_1 \ n_2 \ \cdots \ n_b \ n_{b+1} \ \cdots)_2$$

Next we round the magnitude to b bits, which may be accomplished by adding 2^{-b-1} and then truncating the result to b bits:

$$\begin{array}{r} x = (s . n_1 \ n_2 \ \cdots \ n_b \ n_{b+1} \ n_{b+2} \ \cdots)_{2\text{smms}} \\ + 2^{-b-1} = (0.0 \ 0 \ \cdots \ 0 \ 1 \ 0 \ \cdots)_{2\text{smms}} \\ \hline x + 2^{-b-1} = (s . m_1 \ m_2 \ \cdots \ m_b \ n_{b+1} \oplus 1 \ n_{b+2} \ \cdots)_{2\text{smms}} \\ \qquad \qquad \qquad |Q_r^{b*}x| \end{array} \quad (14-12)$$

Hence, if no overflow occurs,

$$Q_r^b(x) = (s . m_1 \ m_2 \ \cdots \ m_b)_{2\text{smns}} \quad (14-13)$$

Now let us examine the error e_r introduced by the rounding process.

$$e_r = Q_r^b(x) - x$$

First, we may state that

$$x + 2^{-b-1} = (s . m_1 \ m_2 \ \cdots \ m_b \ n_{b+1} \oplus 1 \ n_{b+2} \ n_{b+3} \ \cdots)_2$$

or

$$x + 2^{-b-1} = Q_r^b(x) + 2^{-b}(.n_{b+1} \oplus 1 \ n_{b+2} \ n_{b+3} \ \cdots)_2$$

Thus

$$\underbrace{Q_r^b(x) - x - 2^{-b-1}}_{e_r} = -2^{-b}(.n_{b+1} \oplus 1 \ n_{b+2} \ n_{b+3} \ \cdots)_2 \quad (14-14)$$

But the binary part of the right-hand side is bounded by

$$0 \leq (.n_{b+1} \oplus 1 \ n_{b+2} \ n_{b+3} \ \cdots)_2 < 1 \quad (14-15)$$

and may assume any real number in the interval, so that

$$0 \geq e_r - 2^{-b-1} > -2^{-b} \quad (14-16)$$

Consequently,

$$\frac{2^{-b}}{2} \geq e_r > \frac{-2^{-b}}{2} \quad (14-17)$$

That is, the rounding error e_r is bounded by one-half the least significant bit. The rounding quantizer transfer characteristic is depicted in Figure 14-1b. If the number x may have any number in the range $0 \leq |x| \leq 1$, then e_r may be assumed to have a uniform probability density function as shown in Figure 14-2b. Hence e_r is considered to be a random variable. We may calculate the noise variance as follows:

$$\sigma_{e_r}^2 = E[e_r^2] - E^2[e_r] \quad (14-18)$$

where $E[e_r]$ is the expected value of e_r and

$$E[e_r^k] = \int_{-\infty}^{\infty} e_r^k p(e_r) de_r \quad (14-19)$$

Therefore,

$$\begin{aligned} E[e_r] &= \int_{-2^{-b/2}}^{2^{-b/2}} e_r \cdot 2^b de_r \\ &= \frac{e_r^2}{2} \cdot 2^b \Big|_{-2^{-b/2}}^{2^{-b/2}} = 0 \\ E[e_r^2] &= \int_{-2^{-b/2}}^{2^{-b/2}} e_r^2 \cdot 2^b de_r \\ &= \frac{e_r^3}{3} \cdot 2^b \Big|_{-2^{-b/2}}^{2^{-b/2}} = \frac{2 \cdot 2^b}{3} \left(\frac{2^{-b}}{2} \right)^3 \\ &= \frac{2^{-2b}}{12} \end{aligned} \quad (14-20)$$

Hence

$$\sigma_{e_r}^2 = \frac{2^{-2b}}{12} \quad (14-21)$$

We note that

$$\sigma_{e_t}^2 = \frac{5}{2} \sigma_{e_r}^2 \quad (14-22)$$

Consequently, round-off gives about 1.6 times as much accuracy as truncation using the same number of bits.

Finally, let us examine the *LSB-1* quantization case. This quantizer simply forces the least significant bit to a value 1 in all numbers $Q_l^b(x)$. So if

$$x = (s.m_1 \ m_2 \ \cdots \ m_{b-1} \ m_b \ m_{b+1} \ \cdots)_{2\text{smns}} \quad (14-23)$$

then

$$Q_l^b(x) = (s.m_1 \ m_2 \ \cdots \ m_{b-1} \ 1 \ 0 \ \cdots)_{2\text{smns}} \quad (14-24)$$

Thus the quantizing error is

$$\begin{aligned} e_l &= Q_l^b(x) - x \\ &= 2^{-b+1} \left(\frac{1}{2} - (.m_b \ m_{b+1} \ \cdots)_2 \right) \end{aligned} \quad (14-25)$$

Since

$$0 \leq (.m_b \ m_{b+1} \ \cdots)_2 < 1$$

Then

$$-\frac{1}{2} < \frac{1}{2} - (.m_b \ m_{b+1} \ \cdots)_2 \leq \frac{1}{2} \quad (14-26)$$

Consequently, the error is bounded by

$$-2^{-b} < e_l \leq 2^{-b} \quad (14-27)$$

The quantizer characteristics and probability density function are illustrated in Figures 14-1c and 14-2c, respectively. From Figure 14-2c we may find that the noise variance in this case is

$$\sigma_{e_l}^2 = E[e_l^2] - E^2[e_l] \quad (14-28)$$

But

$$\begin{aligned} E[e_l] &= \int_{-2^{-b}}^2 e_l \left(\frac{2^b}{2} \right) de_l = \frac{2^b}{2} \frac{e_l^2}{2} \Big|_{-2^{-b}}^{2^{-b}} = 0 \\ E[e_l^2] &= \int_{-2^{-b}}^{2^b} e_l^2 \left(\frac{2^b}{2} \right) de_l = \frac{2^b}{2} \frac{e_l^3}{3} \Big|_{-2^{-b}}^{2^{-b}} \\ &= \frac{2^b}{2} \left[\frac{2^{-3b}}{3} + \frac{2^{-3b}}{3} \right] = \frac{2^{-2b}}{3} \end{aligned} \quad (14-29)$$

Hence

$$\sigma_{e_l}^2 = \frac{2^{-2b}}{3} \quad (14-30)$$

and

$$\sigma_{e_l}^2 = 4\sigma_{e_r}^2 \quad (14-31)$$

So, rounding yields twice as much accuracy as LSB-1 quantizing for the same number of bits.

As an interesting exercise, let us examine the relationship between the three quantizers Q_r^b , Q_l^b , and Q_t^b more closely.

In examining the round-off quantizer we actually calculated its error by using the truncation quantizer after adding a term $2^{-b}/2$. Or

$$Q_r^b(x) = Q_t^b \left(x + (1 - 2s) \frac{2^{-b}}{2} \right) \quad (14-32)$$

where the superscript has been added to indicate b bits. We may also express Q_l^b as

$$Q_l^b(x) = Q_t^{b-1}(x) + (1 - 2s)2^{-b} \quad (14-33)$$

A graphical interpretation of these expressions appears in Figure 14-3. In the figure the hexagons indicate a nonlinear transfer characteristic, as shown in Figure 14-1.

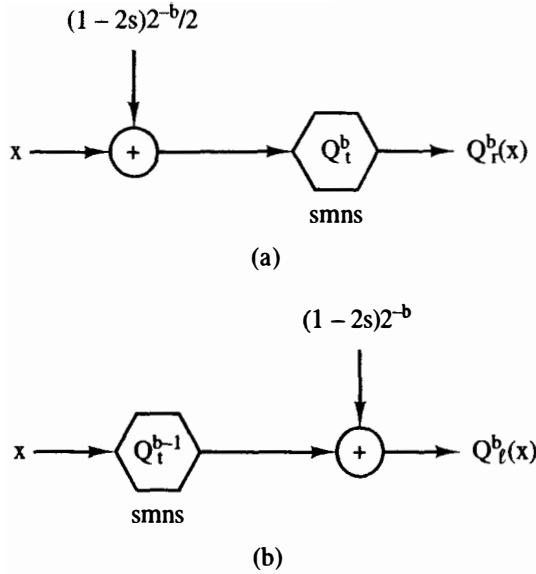


Figure 14-3 Equivalent quantizers: (a) round-off quantizer; (b) LSB-1 quantizer.

Next let us consider the dynamic range of the signed-magnitude number system. If we define *dynamic range* as

$$\text{D.R.} = \frac{\text{largest magnitude}(|Q^b(x)|_{\max})}{\text{smallest nonzero magnitude}(|Q^b(x)|_{\min} \neq 0)} \quad (14-34)$$

then

$$\begin{aligned} (\text{D.R.})_{2\text{smns}} &= \frac{1 - 2^{-b}}{2^{-b}} \\ &= -2^b - 1 \end{aligned} \quad (14-35)$$

and numbers must fall in the range

$$-(2^b - 1) \leq 2^b \cdot Q^b(x) \leq (2^b - 1) \quad (14-36)$$

Suppose that we add two numbers in the binary signed-magnitude number system

$$\begin{array}{r} Q^b(x_1) = (s_1 \cdot m_{11} \quad m_{12} \quad \cdots \quad m_{1b})_{2\text{smns}} \\ + Q^b(x_2) = (s_2 \cdot m_{21} \quad m_{22} \quad \cdots \quad m_{2b})_{2\text{smns}} \\ \hline Q^b(x_3) = (s_3 \cdot m_{31} \quad m_{32} \quad \cdots \quad m_{3b})_{2\text{smns}} \end{array} \quad (14-37)$$

If s_1 and s_2 are different, then

$$\begin{aligned} s_3 &= s_1 & \text{if } |Q^b(x_1)| \geq |Q^b(x_2)| \\ s_3 &= s_2 & \text{if } |Q^b(x_2)| > |Q^b(x_1)| \end{aligned} \quad (14-38)$$

and

$$|Q^b(x_3)| = ||Q^b(x_1)| - |Q^b(x_2)|| \quad (14-39)$$

But if s_1 and s_2 are the same, then

$$s_3 = s_1 = s_2$$

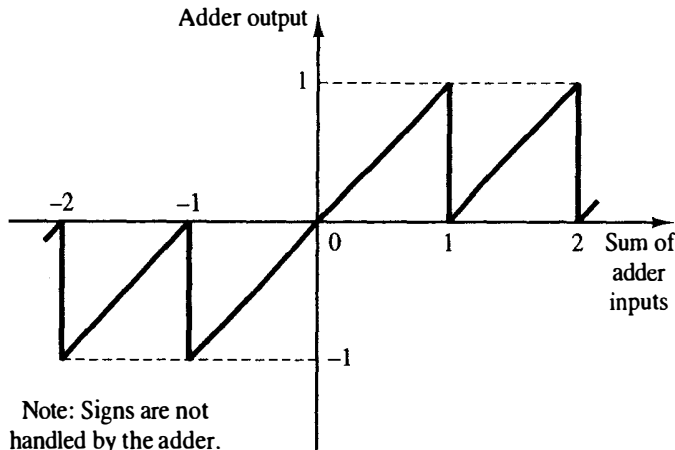


Figure 14-4 Signed-magnitude number system overflow characteristic.

and

$$|Q^b(x_3)| = |Q^b(x_1)| + |Q^b(x_2)| \quad (14-40)$$

and $|Q^b(x_3)|$ may exceed the range of the number system. If the addition of the number magnitudes is accomplished using a binary adder and the overflow bit is ignored, the resulting *overflow* characteristic for the signed-magnitude number system is given in Figure 14-4. This nonlinear overflow characteristic is important in analyzing the closed-loop effects of a digital filter's large-signal behavior in a discrete control system.

Two's-Complement Number System

The two's-complement number system is the number system used in most digital computers, and hence is commonly used to implement digital filters. Numbers are represented as

$$\begin{aligned} Q^b(x) &= (0.m_1 \ m_2 \ \cdots \ m_b)_{2\text{cns}}, & 0 \leq x < 1 \\ &= (1.n_1 \ n_2 \ \cdots \ n_b)_{2\text{cns}}, & -1 \leq x < 0 \end{aligned} \quad (14-41)$$

where, for $x \geq 0$,

$$(.m_1 \ m_2 \ \cdots \ m_b)_2 = |Q^b(x)| = \sum_{i=1}^b m_i \cdot 2^{-i} \quad (14-42)$$

as in the signed-magnitude number system, but for $x < 0$,

$$\begin{aligned} (.n_1 \ n_2 \ \cdots \ n_b)_2 &= \text{two's complement of } |Q(x)| \\ &= 1.0 - |Q^b(x)| \\ &= 1.0 - \sum_{i=1}^b m_i \cdot 2^{-i} \end{aligned} \quad (14-43)$$

In series form

$$\begin{aligned} Q^b(x) &= 2.0 - \sum_{i=0}^b m_i \cdot 2^{-i}, & x < 0 \\ Q^b(x) &= \sum_{i=0}^b m_i \cdot 2^{-i}, & x \geq 0 \end{aligned} \quad (14-44)$$

where m_0 is the sign bit. Here we should emphasize that positive numbers are identical to the signed-magnitude results presented earlier.

First let us consider the action of a truncation quantizer in generating $Q_t^b(x)$. For positive numbers we may use the results of the signed-magnitude analysis.

$$0 \geq e_t > -2^{-b}, \quad x \geq 0 \quad (14-45)$$

For the negative case, however, the result is *not* identical with the signed-magnitude case.

Since

$$\begin{aligned} x &= (1.n_1 \ n_2 \ \cdots \ n_b \ n_{b+1} \ \cdots)_{2\text{cns}} \\ Q_t^b(x) &= (1.n_1 \ n_2 \ \cdots \ n_b \ 0 \ \cdots)_{2\text{cns}} \end{aligned} \quad (14-46)$$

Then

$$Q_t^b(x) - x = -2^{-b}(0.n_{b+1} \ n_{b+2} \ \cdots)_{2\text{cns}}$$

or

$$e_t = -2^{-b} \sum_{i=1}^{\infty} n_{b+i} \cdot 2^{-i} \quad (14-47)$$

Because

$$0 \leq \sum_{i=1}^{\infty} n_{b+i} \cdot 2^{-i} < 1 \quad (14-48)$$

Then

$$0 \geq e_t > -2^{-b}, \quad x < 0 \quad (14-49)$$

which was the result for positive x . Hence for all x ,

$$0 \geq e_t > -2^{-b} \quad (14-50)$$

The quantization characteristic and probability density function for the truncation error are shown in Figures 14-5a and 14-6a, respectively. Notice that the mean value will be nonzero:

$$\begin{aligned} E[e_t] &= \int_{-\infty}^{\infty} e_t p(e_t) de_t \\ &= \int_{-2^{-b}}^0 e_t \cdot 2^b de_t = 2^b \left. \frac{e_t^2}{2} \right|_{-2^{-b}}^0 \\ &= -2^b \frac{2^{-2b}}{2} = \frac{-2^{-b}}{2} \end{aligned} \quad (14-51)$$

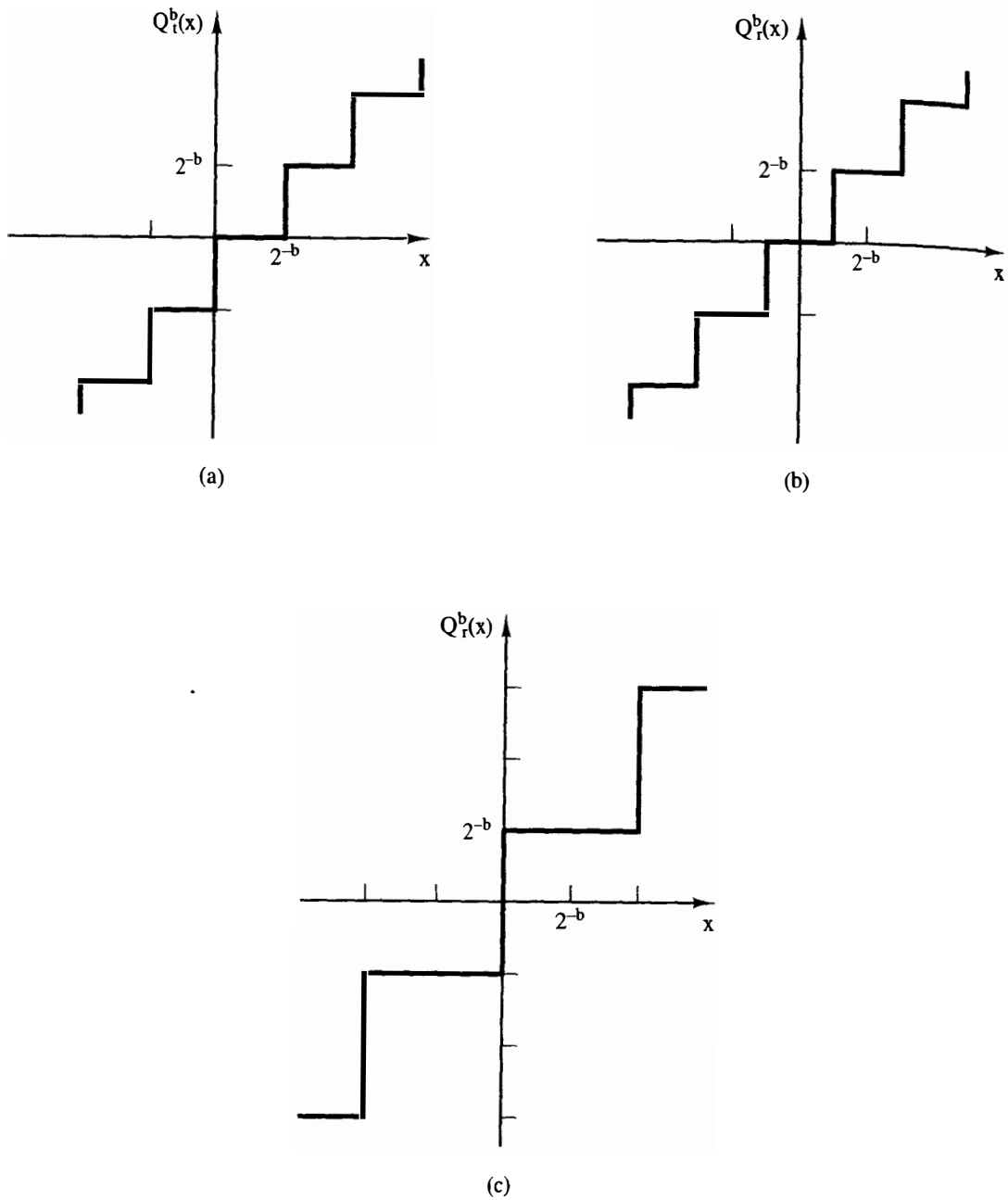


Figure 14-5 Quantizer characteristics for the two's-complement number system: (a) truncation; (b) round-off; (c) LSB-1.

The expected value of e_t^2 is

$$\begin{aligned}
 E[e_t^2] &= \int_{-\infty}^{\infty} e_t^2 p(e_t) de_t \\
 &= \int_{-2^{-b}}^0 e_t^2 \cdot 2^b de_t = 2^b \frac{e_t^3}{3} \Big|_{-2^{-b}}^0 \\
 &= 2^b \frac{2^{-3b}}{3} = \frac{2^{-2b}}{3}
 \end{aligned} \tag{14-52}$$

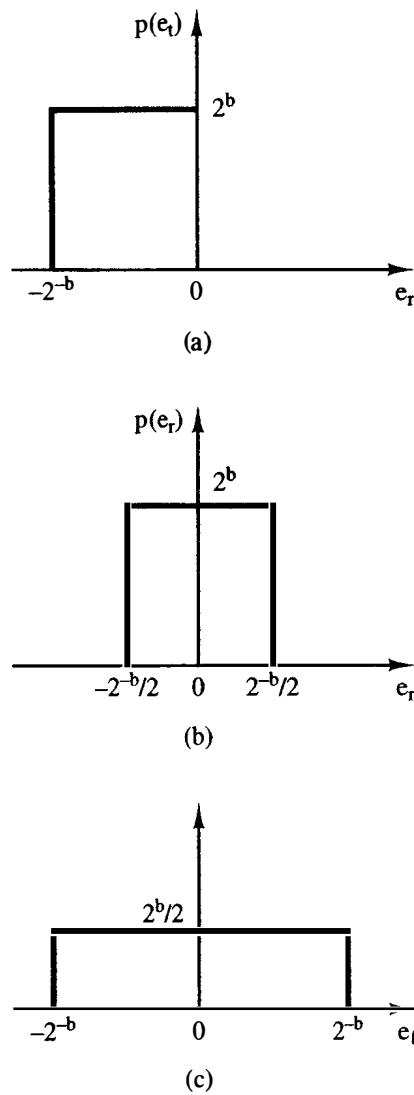


Figure 14-6 Quantizer error probability density functions: (a) truncation; (b) round-off; (c) LSB-1.

And we may calculate the variance

$$\begin{aligned}
 \sigma_{e_t}^2 &= E[e_t^2] - E^2[e_t] \\
 &= \frac{2^{-2b}}{3} - \left(-\frac{2^{-b}}{2}\right)^2 = 2^{-2b} \left(\frac{1}{3} - \frac{1}{4}\right) \\
 &= \frac{2^{-2b}}{12}
 \end{aligned} \tag{14-53}$$

Since two's-complement truncation has a nonzero mean, it can introduce dc biasing errors into the digital filter output. The use of this quantizer in a closed-loop control system requires that its effect be analyzed carefully in the closed-loop case.

Now suppose that we consider the case of the round-off quantizer in determining $Q_r^b(x)$. For positive numbers, round-off in the two's-complement number system duplicates the signed-magnitude case because the numbers have the same representation. Hence, from the signed-magnitude case, we have

$$\frac{2^{-b}}{2} \geq e_r > -\frac{2^{-b}}{2} \tag{14-54}$$

For the negative case, let

$$x = 2.0 - \sum_{i=0}^{\infty} n_i \cdot 2^{-i} \quad (14-55)$$

Recalling our development for the signed-magnitude case,

$$\begin{array}{r} x = (1 . n_1 \ n_2 \ \cdots \ n_b \ n_{b+1} \ n_{b+2} \ \cdots)_{2\text{cns}} \\ + 2^{-b-1} = (0.0 \ 0 \ \cdots \ 0 \ 1 \ 0 \ \cdots)_{2\text{cns}} \\ \hline x + 2^{-b-1} = \underbrace{(1 . k_1 \ k_2 \ \cdots \ k_b \ n_{b+1} \oplus 1 \ n_{b+2} \ \cdots)_{2\text{cns}}}_{Q_r^b(x)} \end{array} \quad (14-56)$$

and

$$x + 2^{-b-1} = Q_r^b(x) + 2^{-b}(0 . n_{b+1} \oplus 1 \ n_{b+2} \ \cdots)_{2\text{cns}} \quad (14-57)$$

But

$$0 \leq (0 . n_{b+1} \oplus 1 \ n_{b+2} \ \cdots)_{2\text{cns}} < 1 \quad (14-58)$$

then

$$\frac{2^{-b}}{2} \geq e_r > -\frac{2^{-b}}{2}, \quad x < 0 \quad (14-59)$$

Consequently, round-off for the two's-complement number system is the same as for signed-magnitude numbers, and the quantizer characteristic of Figure 14-5b holds. Therefore, the error distribution may be assumed to have the form of Figure 14-6b and

$$\sigma_{e_r}^2 = \frac{2^{-2b}}{12} \quad (14-60)$$

Finally, let us examine the LSB-1 quantizer for the two's-complement number system. Positive results are the same as the signed-magnitude case. Since negative numbers are expressed

$$x = (1 . n_1 n_2 \ \cdots \ n_{b-1} \ n_b \ n_{b+1} \ \cdots)_{2\text{cns}}, \quad x < 0 \quad (14-61)$$

Then

$$\begin{aligned} x &= -\left(2.0 - \sum_{i=0}^{\infty} n_i \cdot 2^{-i}\right) \\ Q_l^b(x) &= -\left(2.0 - 2^{-b} - \sum_{i=0}^{b-1} n_i \cdot 2^{-i}\right) \end{aligned} \quad (14-62)$$

Hence

$$\begin{aligned} e_l &= -2.0 + 2^{-b} + \sum_{i=0}^{b-1} n_i \cdot 2^{-i} + 2.0 - \sum_{i=0}^{\infty} n_i \cdot 2^{-i} \\ &= 2^{-b} - \sum_{i=b}^{\infty} n_i \cdot 2^{-i} \end{aligned} \quad (14-63)$$

But if we let $k = i - b$,

$$\begin{aligned} e_i &= 2^{-b} - \sum_{k=0}^{\infty} n_{b+k} \cdot 2^{-b-k} \\ &= 2^{-b} \left(1 - \sum_{k=0}^{\infty} n_{b+k} \cdot 2^{-k} \right) \end{aligned} \quad (14-64)$$

Since

$$0 \leq \sum_{k=0}^{\infty} n_{b+k} \cdot 2^{-k} < 2.0 \quad (14-65)$$

then

$$-1 < 1 - \sum_{k=0}^{\infty} n_{b+k} \cdot 2^{-k} \leq 1 \quad (14-66)$$

so that

$$-2^{-b} < e_i \leq 2^{-b}, \quad x < 0 \quad (14-67)$$

which was our result for positive x and the relation thus holds for all x .

The quantizer characteristic is demonstrated in Figure 14-5c and the probability density function for e_i is plotted in Figure 14-6c. These curves were the same ones for the signed-magnitude case, so

$$\sigma_{e_i}^2 = \frac{2^{-2b}}{3} \quad (14-68)$$

We may compare the three quantizers by examining Figure 14-7. Here we have used the truncation quantizer to implement round-off and LSB-1 quantizers, because

$$Q_r^b(x) = Q_t^b \left(x + \frac{2^{-b}}{2} \right) \quad (14-69)$$

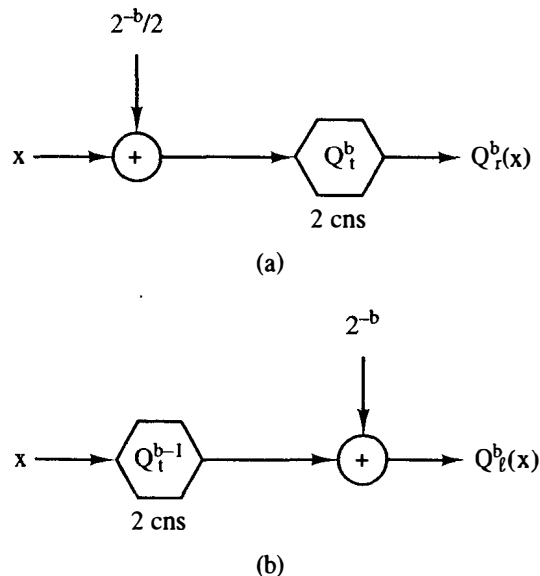


Figure 14-7 Equivalent quantizers for the two's-complement number system: (a) round-off quantizer; (b) LSB-1 quantizer.

and

$$Q_i^b(x) = Q_i^{b-1}(x) + 2^{-b} \quad (14-70)$$

Consider next the dynamic range of the two's-complement number system. Since

$$-1 \leq Q^b(x) \leq 1 - 2^{-b} \quad (14-71)$$

the dynamic range is given by

$$\begin{aligned} (\text{D.R.})_{2\text{cns}} &= \frac{|Q^b(x)|_{\max}}{|Q^b(x)|_{\min} \neq 0} \\ &= \frac{|-1|}{|2^{-b}|} \\ &= 2^b \end{aligned} \quad (14-72)$$

Finally, let us examine the overflow properties of the two's-complement number system. If we add two numbers to generate a third:

$$\begin{array}{r} Q^b(x_1) = (m_{10} . m_{11} \quad m_{12} \quad \cdots \quad m_{1b})_{2\text{cns}} \\ + Q^b(x_2) = (m_{20} . m_{21} \quad m_{22} \quad \cdots \quad m_{2b})_{2\text{cns}} \\ \hline Q^b(x_3) = (m_{30} . m_{31} \quad m_{32} \quad \cdots \quad m_{3b})_{2\text{cns}} \end{array} \quad (14-73)$$

↑
sign position

the binary sum is calculated and the signs are automatically handled by the adder circuits. Any carry bit into the 2^1 position is ignored; this happens when two negative numbers are added. Overflow occurs when two positive numbers are added and a carry bit enters the sign position (2^0 position), or when two negative numbers are added and the carry bit into the sign position is absent. Hence the overflow characteristic for the two's-complement number system is shown in Figure 14-8.

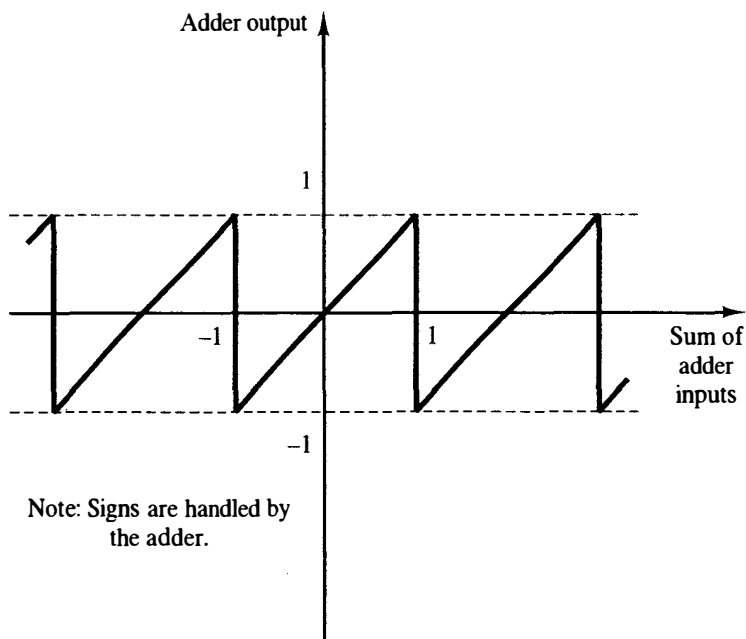


Figure 14-8 Two's-complement number system overflow characteristic.

14.3 COEFFICIENT QUANTIZATION

One effect of finite wordlengths in digital computers is that the filter's parameters, or coefficients, must be chosen from a finite set of allowable values. Classical design procedures yield filter transfer functions with coefficients of arbitrary precision which must be altered for implementation using digital computing devices. One approach to this problem is to select a filter structure that is not sensitive to coefficient inaccuracies. For example, realizing a filter directly allows a greater chance for instability than cascading or paralleling second-order modules because it is well known that the roots of polynomials become more sensitive to parameter changes as the order of the polynomial increases.

Pole—Zero Locations

Quantizing the coefficients of a digital filter effectively restricts the poles and zeros of the filter to lie on a finite number of discrete points in the z -plane. Consider the second-order filter

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \quad (14-74)$$

implemented in the 1D structure (see Figure 14-9a). Since zeros cannot cause instability, let us examine only the poles. The poles are given

$$\begin{aligned} (z - re^{j\theta})(z - re^{-j\theta}) &= z^2 - (2r \cos \theta)z + r^2 \\ &= z^2 + b_1 z + b_2 \end{aligned} \quad (14-75)$$

Hence

$$b_1 = -2r \cos \theta, \quad b_2 = r^2$$

Suppose that we quantize b_1 to b_1^q and b_2 to b_2^q . Figure 14-9b illustrates that quantizing b_1 (i.e., $2r \cos \theta$) restricts the poles to lie on a finite number of vertical lines, $r \cos \theta = b_1^q/2$, while quantizing b_2 (i.e., r^2) further restricts the poles to lie on circles of radius $r = \sqrt{b_2^q}$.

Compare the 1D structure with the 1X structure of Figure 14-9c. The coefficients g_i are defined in Chapter 12. The poles are defined by

$$g_1 = r \cos \theta$$

$$g_2 = r \sin \theta$$

If we quantize g_1 to g_1^q and g_2 to g_2^q , then

$$Q(r \cos \theta) = g_1^q$$

and

$$Q(r \sin \theta) = g_2^q$$

restricts the poles to lie on a rectangular grid.

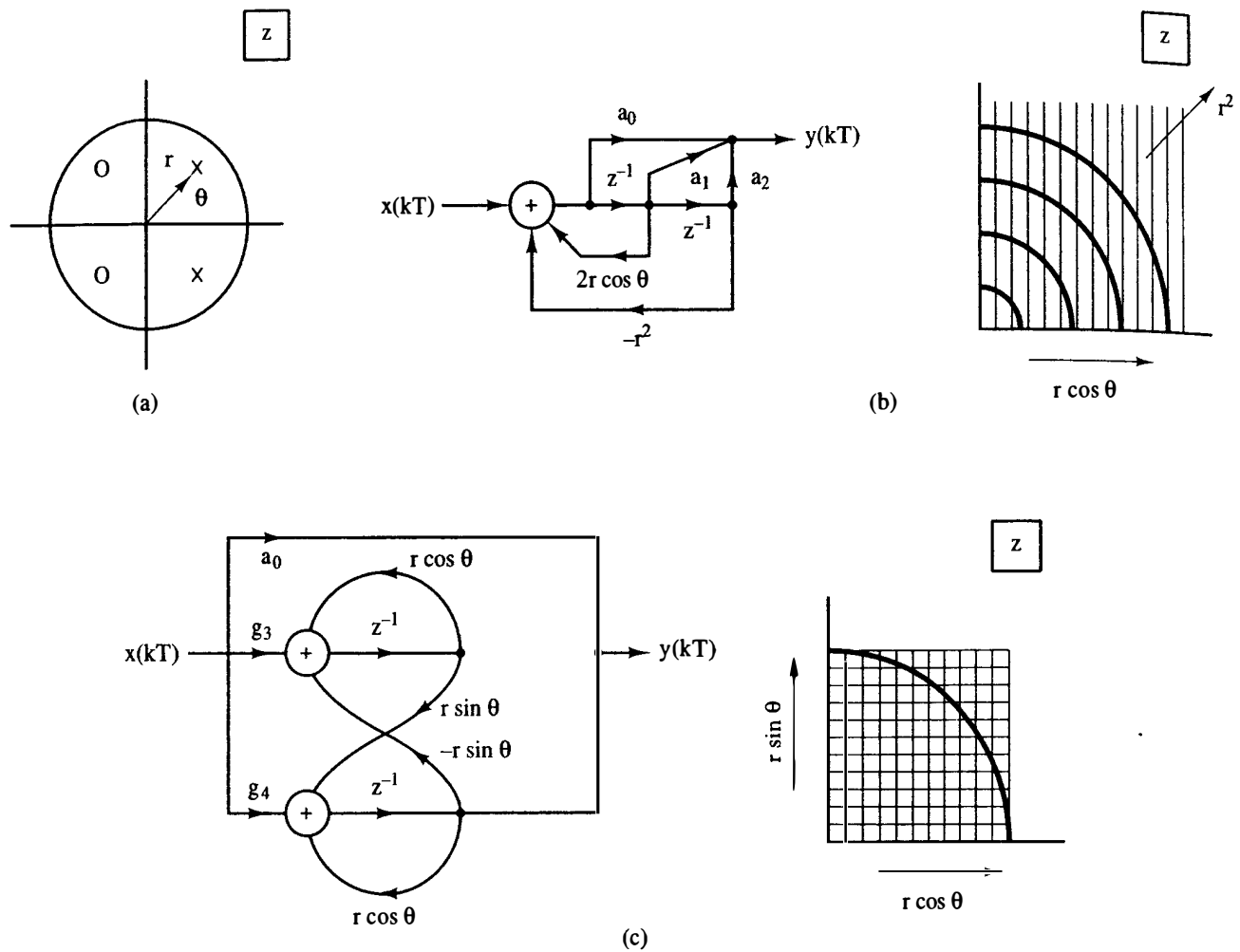


Figure 14-9 Coefficient quantization: (a) z -plane; (b) 1D structure; (c) 1X structure.

Consequently, one can see that the 1D structure is better suited to implement digital filters with poles near the unit circle, whereas the 1X structure gives a more uniform pattern of realizable locations throughout the unit circle.

Error Analysis

Sensitivity analysis may be employed to determine the effect of coefficient quantization on either the pole migration or change in transfer characteristic. Let us examine pole migration in the 1D filter structure. From Figure 14-9,

$$\begin{aligned} b_1 &= -2r \cos \theta \\ b_2 &= r^2 \end{aligned} \quad (14-76)$$

The pole migration is given by

$$\begin{aligned}\Delta r &= \frac{\partial r}{\partial b_1} \Delta b_1 + \frac{\partial r}{\partial b_2} \Delta b_2 \\ \Delta \theta &= \frac{\partial \theta}{\partial b_1} \Delta b_1 + \frac{\partial \theta}{\partial b_2} \Delta b_2\end{aligned}\quad (14-77)$$

Using the relationship above gives

$$\begin{aligned}\frac{\partial r}{\partial b_1} &= -\frac{1}{2 \cos \theta} \\ \frac{\partial r}{\partial b_2} &= \frac{1}{2r} \\ \frac{\partial \theta}{\partial b_1} &= \frac{1}{2r \sin \theta} \\ \frac{\partial \theta}{\partial b_2} &= \frac{1}{2r^2 \tan \theta}\end{aligned}\quad (14-78)$$

Consequently,

$$\begin{aligned}\Delta r &= \frac{-\Delta b_1}{2 \cos \theta} + \frac{\Delta b_2}{2r} \\ \Delta \theta &= \frac{\Delta b_1}{2r \sin \theta} + \frac{\Delta b_2}{2r^2 \tan \theta}\end{aligned}\quad (14-79)$$

These relations show us that, for a given r , as θ goes to zero, $\Delta \theta$ approaches infinity. Similarly for a constant θ , as r goes to zero, $\Delta \theta$ and Δr both go to infinity. These results agree with the grid pattern depicted in Figure 14-9b.

Consider now the effect of coefficient quantization on the transfer function of the digital filter [1] (consider Figure 14-10). The point a is the input node for the filter; point b , the output node. Points n and m are two arbitrary internal nodes and $F_{nm}(z)$ is the transfer function of a branch of the network from node n to node m . If we define $T_{ij}(z)$ to be the transfer function between any two network nodes i and j , then the sensitivity of $D(z)$ with respect to $F_{nm}(z)$ is given by

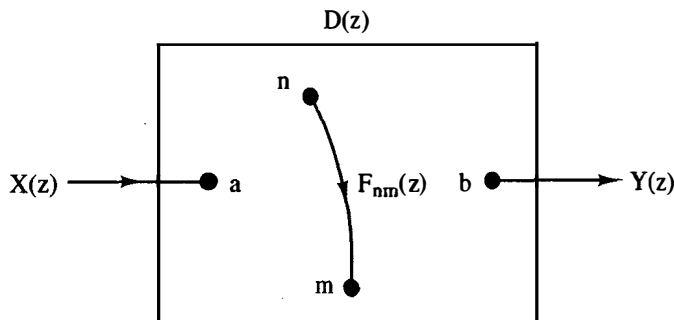


Figure 14-10 Generalized filter.

$$\frac{\partial D(z)}{\partial F_{nm}(z)} = T_{an}(z)T_{mb}(z) \quad (14-80)$$

the product of two transfer functions within the network [2]. If the network branch is a multiplicative coefficient

$$F_{nm}(z) = f_{cnm}$$

then

$$\frac{\partial |D(z)|}{\partial f_{cnm}} = \operatorname{Re} \left[\frac{|D(z)|}{D(z)} T_{an}(z) T_{mb}(z) \right] \quad (14-81)$$

If however, the branch includes a time-delay element

$$F_{nm}(z) = f_{dnm} z^{-1}$$

then

$$\frac{\partial |D(z)|}{\partial f_{dnm}} = \operatorname{Re} \left[\frac{|D(z)|}{D(z)} T_{an}(z) T_{mb}(z) z^{-1} \right] \quad (14-82)$$

Both f_{cnm} and f_{dnm} are real coefficients.

The sensitivities above may be used to calculate the change in the filter transfer function $D(z)$ due to a change in coefficient value. Consider the 1X structure of Figure 14-11. We may calculate the sensitivities mathematically:

$$\begin{aligned} D(z) &= \frac{a_0 + g_4 z^{-1} - g_2 g_3 z^{-2}}{1 - 2g_1 z^{-1} + (g_1 + g_2) z^{-2}} \\ &= a_0 + \frac{(g_4 - jg_3)/2}{z - g_1 + jg_2} + \frac{(g_4 + jg_3)/2}{z - g_1 - jg_2} \end{aligned} \quad (14-83)$$

and

$$\begin{aligned} \frac{\partial D(z)}{\partial a_0} &= 1 \\ \frac{\partial D(z)}{\partial g_1} &= \frac{-(g_4 - jg_3)/2}{(z - g_1 + jg_2)^2} + \frac{-(g_4 + jg_3)/2}{(z - g_1 - jg_2)^2} \\ \frac{\partial D(z)}{\partial g_2} &= \frac{-j(g_4 - jg_3)/2}{(z - g_1 + jg_2)^2} + \frac{-j(g_4 + jg_3)/2}{(z - g_1 - jg_2)^2} \\ \frac{\partial D(z)}{\partial g_3} &= \frac{-j/2}{z - g_1 + jg_2} + \frac{j/2}{z - g_1 - jg_2} = \frac{g_2}{(z - g_1)^2 + g_2^2} \\ \frac{\partial D(z)}{\partial g_4} &= \frac{1/2}{z - g_1 + jg_2} + \frac{1/2}{z - g_1 - jg_2} = \frac{z - g_1}{(z - g_1)^2 + g_2^2} \end{aligned} \quad (14-84)$$

Finally,

$$\Delta D(z) = \frac{\partial D(z)}{\partial a_0} \Delta a_0 + \sum_{i=1}^4 \frac{\partial D(z)}{\partial g_i} \Delta g_i \quad (14-85)$$

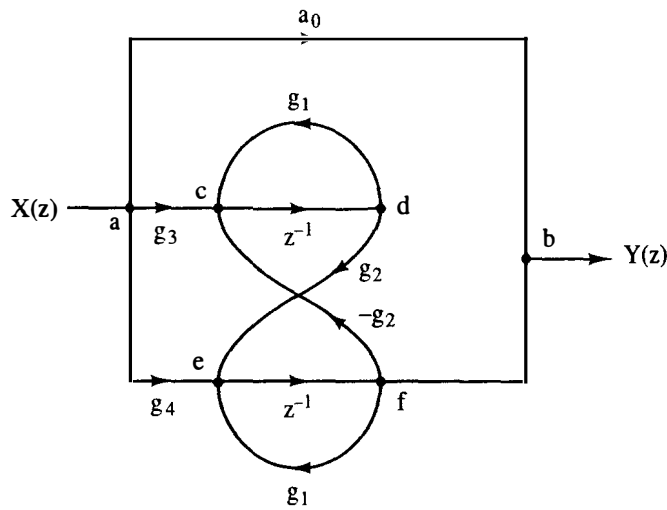


Figure 14-11 1X structure.

Recall that we may calculate the sensitivities using transfer functions and (14-80). For example, from Figure 14-11,

$$\frac{\partial D(z)}{\partial g_3} = T_{aa}(z)T_{cb}(z) \quad (14-86)$$

But $T_{aa}(z) = 1$. However, using Mason's gain formula,

$$\begin{aligned} T_{cb}(z) &= \frac{g_2 z^{-2}}{1 - 2g_1 z^{-1} + g_2^2 z^{-2} + g_1^2 z^{-2}} \\ &= \frac{g_2 z^{-2}}{1 - 2g_1 z^{-1} + g_1^2 z^{-2} + g_2^2 z^{-2}} \\ &= \frac{g_2}{(z - g_1)^2 + g_2^2} \end{aligned} \quad (14-87)$$

Hence

$$\frac{\partial D(z)}{\partial g_3} = \frac{g_2}{(z - g_1)^2 + g_2^2}$$

which confirms our previous result in (14-84).

Observations

When digital filters are used in feedback control systems, the quantization of the filter coefficients can dramatically affect the system closed-loop performance in applications where pole and/or zero placement is critical. If the controller designer employs quantized coefficient values as he or she originally develops the digital filter transfer function, coefficient sensitivity problems may be avoided from the beginning of the design process.

When a digital filter must be quantized for some reason (say, that a change in filter structure is desirable), the quantized filter should be returned to the controller

designers to confirm that the resulting closed-loop characteristics (gain margin, phase margin, time response, etc.) are still within system specifications. The authors have found that, in most applications, coefficient quantization is rarely a problem.

14.4 SIGNAL QUANTIZATION ANALYSIS

In the preceding section we examined the effect of quantizing the coefficients of the filter transfer function. In this section we examine the quantization of the digital filter's signal variables, both at the input and internal nodes.

Filter Input Quantization

The input signal to a digital filter may come from an analog-to-digital (A/D) converter or from the output node of some other digital filter module. Consider first the case of the A/D as illustrated in Figure 14-12. The signal $x(t)$ is sampled and quantized into a sequence of time samples $\{Q(x(n))\}$ which are processed by the

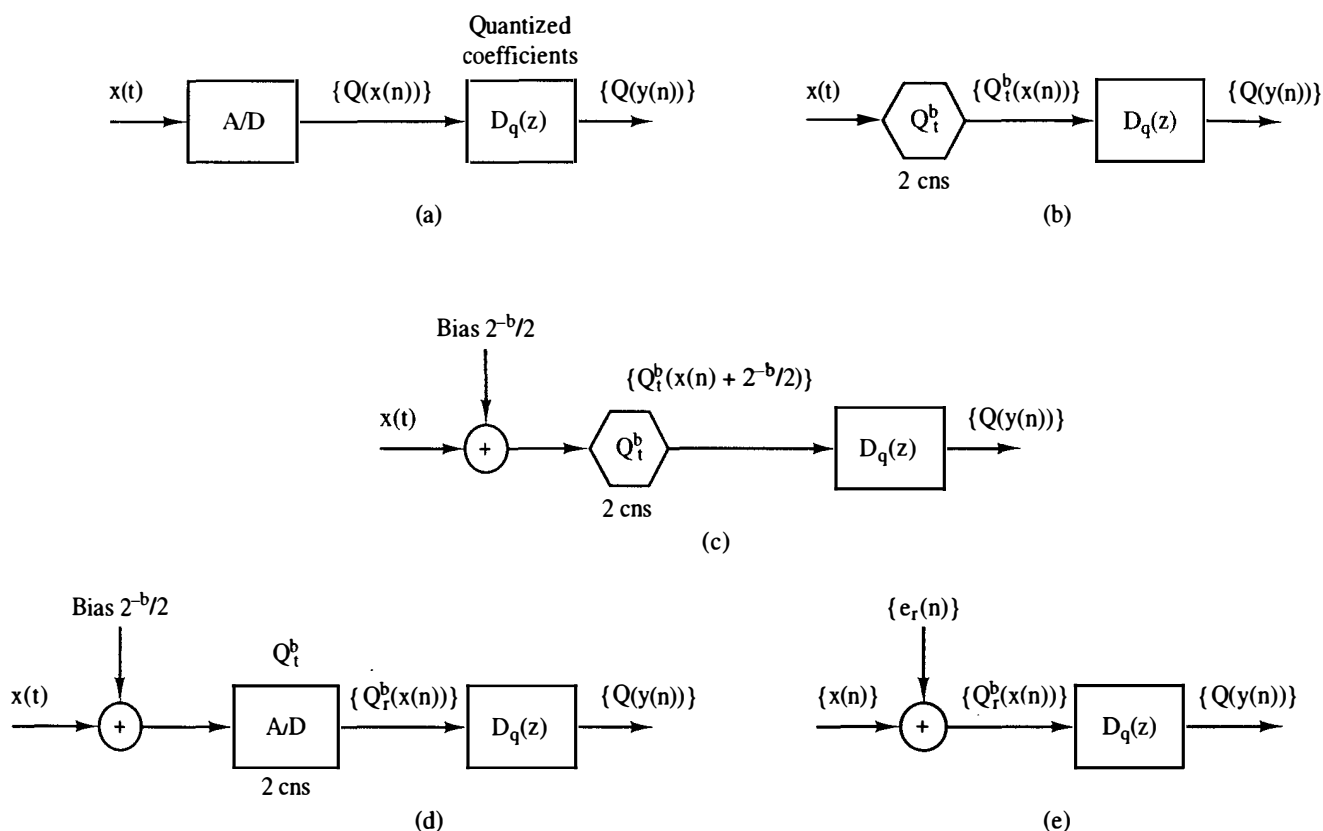


Figure 14-12 (a) General A/D; (b) successive-approximation A/D; (c) biased successive approximation A/D; (d) A/D model; (e) noise model.

digital filter. Perhaps the most common A/D type is the bipolar (positive and negative values) successive-approximation converter, whose conversion time is proportional to the number of bits, $(b + 1)$. This converter is a truncation-type quantizer (see Figure 14-5a) whose output is in the two's-complement number system, so we may represent the A/D as shown in Figure 14-12b. If we bias the input by a small signal $2^{-b}/2$, we form the configuration of Figure 14-12c which, by Figure 14-7a, is equivalent to rounding the input values (Figure 14-12d). This is the A/D model that we use in our results throughout this chapter. The A/D conversion introduces round-off noise into the digital filter as modeled in Figure 14-12e, where

$$e_r(n) = Q_r^b(x(n)) - x(n) \quad (14-88)$$

is the value of the round-off noise at time nT . The round-off noise is assumed to be uniformly distributed as shown in Figure 14-6b, with a variance of

$$\sigma_{e_r}^2 = \frac{2^{-2b}}{12} \quad (14-89)$$

We also see that the noise is bounded by

$$\frac{2^{-b}}{2} \geq e_r > -\frac{2^{-b}}{2} \quad (14-90)$$

Another important point must be made about the successive-approximation A/D of Figure 14-12d. This A/D usually exhibits saturation when the input signal $x(t)$ exceeds its dynamic range. Its overflow characteristic differs from the two's-complement addition characteristic of Figure 14-8. The A/D overflow characteristic is modeled in Figure 14-13.

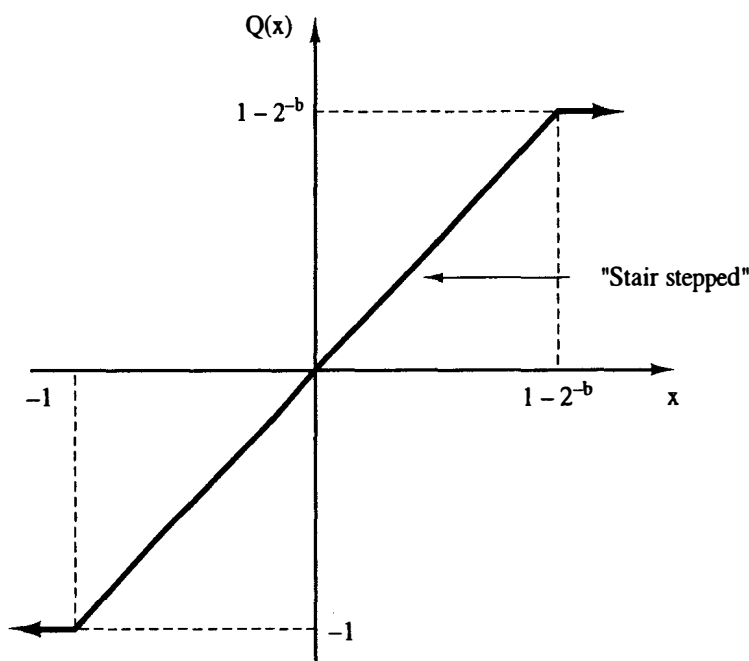


Figure 14-13 A/D overflow.

Internal Variable Quantization

In Chapter 12 we examined many different structures for digital filters. The internal nodes were always formed by summing product terms, with each product being generated by a filter coefficient and a signal variable. If $v_i(n)$ represents the internal variables and c_i the filter coefficients, an internal node $v_k(n)$ is generated by

$$v_k(n) = \sum_{i=1}^L c_i v_i(n) \quad (14-91)$$

the sum of several, say L , product terms. The process is illustrated in Figure 14-14a. This represents the ideal case with no quantization. If, however, the coefficients (c_i) are quantized (c_i^q) as discussed in the preceding section, and if the variables are represented in a finite-wordlength number system, say two's complement, the physically realizable case of Figure 14-14b results. Note that if the coefficients are quantized to a bits and the variables are quantized to b bits, the product terms $c_i^q Q^b(v_i(n))$ will have $a + b$ bits. The product terms may then be quantized to b bits by the quantizers labeled Q_1 , or they may be summed in their entirety ($a + b$ bits) and the resulting sum quantized to b bits by quantizer Q_2 . The choice of quantizing at location Q_1 versus location Q_2 must be made by evaluating the hardware required to compute $a + b$ bits versus the improved quantization noise performance. The noise model is depicted in Figure 14-14c. Note we have simplified the notation setting $v_i^q(n) = Q^b(v_i(n))$. Assuming that

$$\sigma_{e_1}^2 = \sigma_{e_2}^2 \quad (14-92)$$

quantization at point Q_1 generates a noise variance, σ_e^2 , at $v_k^l(n)$ of

$$\sigma_e^2 = \sum_{i=1}^L \sigma_{e_1}^2 = L \sigma_{e_1}^2 = L \sigma_{e_2}^2 \quad (14-93)$$

whereas if quantization is delayed until point Q_2 ,

$$\sigma_e^2 = \sigma_{e_2}^2 \quad (14-94)$$

In other words, quantization at point Q_1 is L times as noisy as point Q_2 . The final noise model is shown in Figure 14-14d.

Now let us examine the nature of the quantization error distributions for the error sources e_1 and e_2 . Examine the product

$$\begin{aligned} v_i^q &= (m_{10} . m_{11} \quad m_{12} \quad \cdots \quad m_{1b})_{2\text{cns}} \\ \times c_i^q &= (m_{20} . m_{21} \quad m_{22} \quad \cdots \quad m_{2a})_{2\text{cns}} \\ \hline c_i^q v_i^q &= (m_{30} . m_{31} \quad m_{32} \quad \cdots \quad m_{3b} \quad m_{3,b+1} \quad \cdots \quad m_{3,b+a})_{2\text{cns}} \end{aligned} \quad (14-95)$$

When this product is quantized (say round-off is used) to b bits, then

$$Q_r^b(c_i^q v_i^q) = (m_{40} . m_{41} \quad m_{42} \quad \cdots \quad m_{4b})_{2\text{cns}} \quad (14-96)$$

the resulting round-off error may be expressed as

$$e_r = Q_r^b(c_i^q v_i^q) - c_i^q v_i^q \quad (14-97)$$

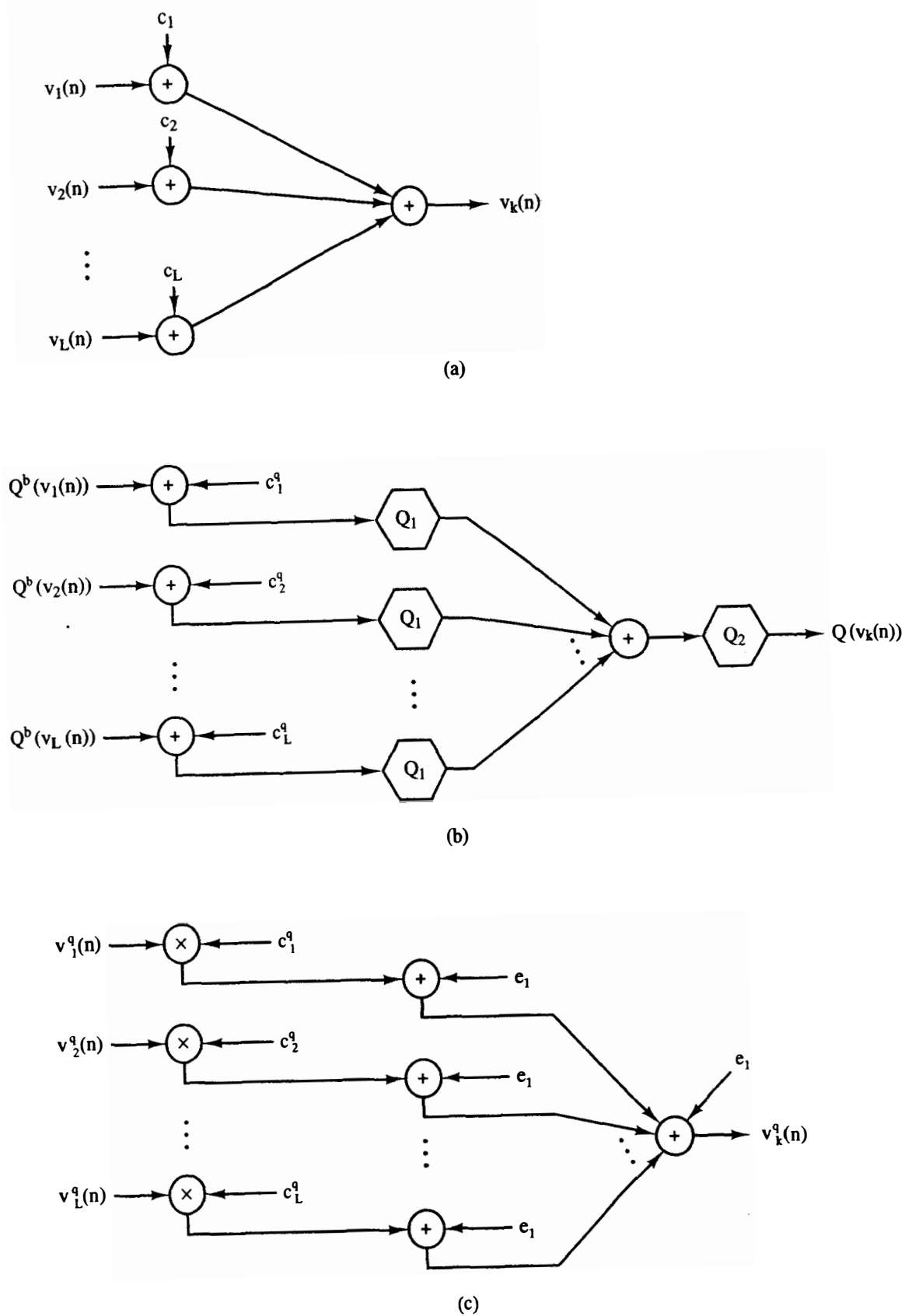


Figure 14-14 (a) Ideal case; (b) physically realizable case; (c) quantization noise model; (d) equivalent noise model.

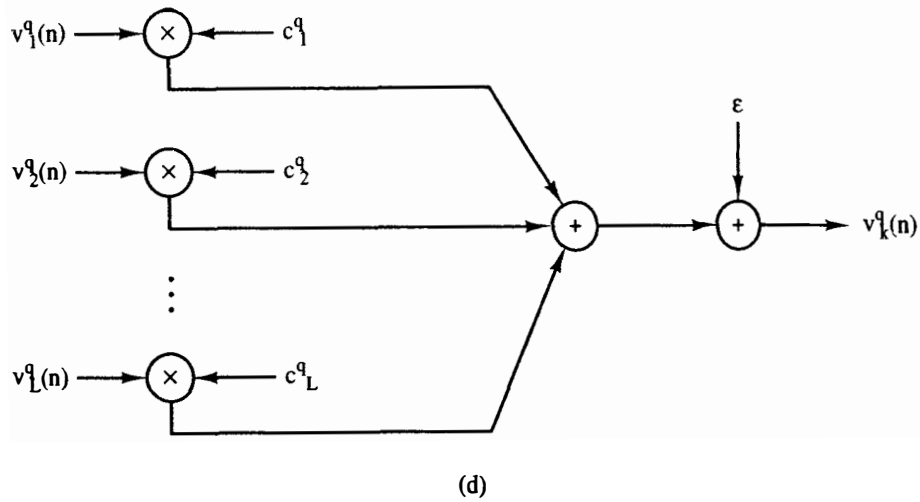


Figure 14-14 (continued)

But

$$\begin{aligned}
 c_i^q v_i^q &= (m_{30} \cdot m_{31} \quad m_{32} \quad \cdots \quad m_{3b} \quad m_{3,b+1} \quad m_{3,b+2} \quad \cdots \quad m_{3,b+a})_{2\text{cns}} \\
 + 2^{-b-1} &= (0 \quad .0 \quad 0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0)_{2\text{cns}} \\
 \hline
 2^{-b-1} + c_i^q v_i^q &= (\underbrace{m_{40} \cdot m_{41} \quad m_{42} \quad \cdots \quad m_{4b}}_{Q_r^b(c_i^q v_i^q)} \quad m_{3,b+1} \oplus 1 \quad m_{3,b+2} \quad \cdots \quad m_{3,b+a})_{2\text{cns}}
 \end{aligned} \tag{14-98}$$

Hence

$$\begin{aligned}
 e_r &= 2^{-b-1} - (0.00 \quad \cdots \quad 0 \quad m_{3,b+1} \oplus 1 \quad m_{3,b+2} \quad \cdots \quad m_{3,b+a})_{2\text{cns}} \\
 &= 2^{-b}(2^{-1} - (.m_{3,b+1} \oplus 1 \quad m_{3,b+2} \quad \cdots \quad m_{3,b+a})_2)
 \end{aligned} \tag{14-99}$$

The binary number may be simplified by letting $m_1 = m_{3,b+1} \oplus 1$ and $m_i = m_{3,b+i}$ for $i = 2, a$, yielding

$$e_r = 2^{-b}(\frac{1}{2} - (.m_1 \quad m_2 \quad \cdots \quad m_a)_2) \tag{14-100}$$

Let us examine the binary number further. It is represented by a bits and may take on discrete value in the range

$$0 \leq (.m_1 \quad m_2 \quad \cdots \quad m_a)_2 \leq (1 - 2^{-a}) \tag{14-101}$$

Then

$$\begin{aligned}
 0 &\geq -(.m_1 \quad m_2 \quad \cdots \quad m_a)_2 \geq 1 - 2^{-a} \\
 \frac{1}{2} &\geq \frac{1}{2} - (.m_1 \quad m_2 \quad \cdots \quad m_a)_2 \geq -(\frac{1}{2} - 2^{-a})
 \end{aligned}$$

and hence the bounds on the error are

$$\frac{2^{-b}}{2} \geq e_r \geq -\frac{2^{-b}}{2} + 2^{-b-a} \tag{14-102}$$

But e_r may be expressed

$$\begin{aligned} 2^b e_r &= (.10 \cdots 0)_2 - (.m_1 m_2 \cdots m_a)_2 \\ &= (n_0 .n_1 n_2 \cdots n_a)_{2\text{cns}} \end{aligned} \quad (14-103)$$

and hence its smallest nonzero magnitude is

$$|e_r|_{\min \neq 0} = 2^{-b-a} \quad (14-104)$$

and its probability density function will be discrete and appear as illustrated in Figure 14-15. The density function consists of a series of a different impulse functions, each of weight 2^{-a} .

The probability density function may be expressed as

$$p(e_r) = 2^{-a} \sum_{i=0}^{2^a-1} \delta[e_r - 2^{-b-1}(1 - i2^{-a+1})] \quad (14-105)$$

where $\delta(x)$ is the Dirac delta function. The expected value of e_r is thus

$$\begin{aligned} E[e_r] &= \int_{-\infty}^{\infty} e_r p(e_r) de_r \\ &= \int_{-\infty}^{\infty} e_r \left\{ 2^{-a} \sum_{i=0}^{2^a-1} \delta[e_r - 2^{-b-1}(1 - i2^{-a+1})] \right\} de_r \\ &= 2^{-a} \sum_{i=0}^{2^a-1} \int_{-\infty}^{\infty} e_r \delta[e_r - 2^{-b-1}(1 - i2^{-a+1})] de_r \end{aligned} \quad (14-106)$$

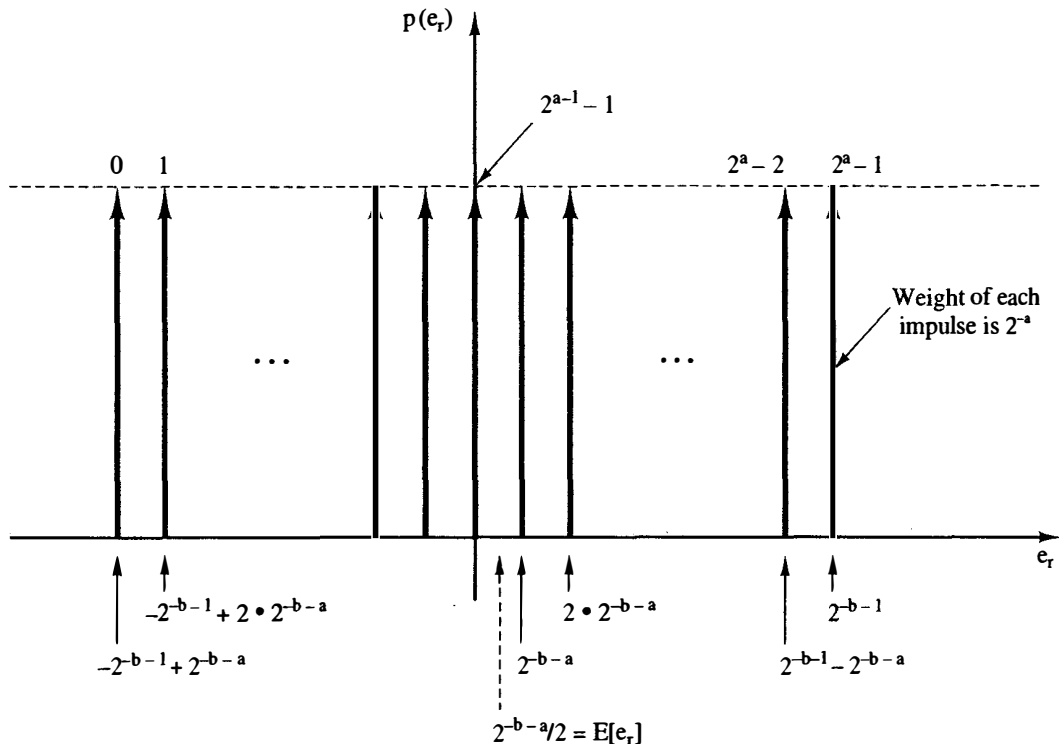


Figure 14-15 Round-off error probability density function.

But we know that

$$\int_{-\infty}^{\infty} x^r \delta(x - y) dx = y^r \quad (14-107)$$

so

$$\begin{aligned} E[e_r] &= 2^{-a} \sum_{i=0}^{2^a-1} [2^{-b-1}(1 - i2^{-a+1})] \\ &= 2^{-a-b-1} \left[\sum_{i=0}^{2^a-1} 1 - \sum_{i=0}^{2^a-1} i2^{-a+1} \right] \end{aligned} \quad (14-108)$$

But the sum of integers is

$$\begin{aligned} \sum_{i=0}^n 1 &= n + 1 \\ \sum_{i=0}^n i &= \frac{n(n+1)}{2} \end{aligned} \quad (14-109)$$

Hence

$$\begin{aligned} E[e_r] &= 2^{-a-b-1} \frac{2^a - 2^{-a+1}(2^a - 1)(2^a)}{2} \\ &= 2^{-a-b-1} [2^a - 2^a + 1] \\ &= \frac{2^{-b-a}}{2} \end{aligned} \quad (14-110)$$

Note this value is indicated on Figure 14-15.

Next let us calculate the expected value of e_r^2 .

$$\begin{aligned} E[e_r^2] &= \int_{-\infty}^{\infty} e_r^2 p(e_r) de_r \\ &= \int_{-\infty}^{\infty} e_r^2 \left\{ 2^{-a} \sum_{i=0}^{2^a-1} \delta[e_r - 2^{-b-1}(1 - i2^{-a+1})] \right\} de_r \\ &= 2^{-a} \sum_{i=0}^{2^a-1} \int_{-\infty}^{\infty} e_r^2 \delta[e_r - 2^{-b-1}(1 - i2^{-a+1})] de_r \\ &= 2^{-a} \sum_{i=0}^{2^a-1} [2^{-b-1}(1 - i2^{-a+1})]^2 \\ &= 2^{-a} 2^{-2b-2} \sum_{i=0}^{2^a-1} [1 - 2i2^{-a+1} + i^2 2^{-2a+2}] \\ &= 2^{-a-2b-2} \left[\left(\sum_{i=0}^{2^a-1} 1 \right) - 2^{-a+2} \left(\sum_{i=0}^{2^a-1} i \right) + 2^{-2a+2} \left(\sum_{i=0}^{2^a-1} i^2 \right) \right] \end{aligned} \quad (14-111)$$

But the sum

$$\sum_{i=0}^n i^2 = \frac{n(n+1)(2n+1)}{6} \quad (14-112)$$

Consequently,

$$\begin{aligned} E[e_r^2] &= 2^{-a-2b-2} \left[2^a - \frac{2^{-a+2}(2^a-1)(2^a)}{2} + \frac{2^{-2a+2}(2^a-1)(2^a)(2^{a+1}-2+1)}{6} \right] \\ &= 2^{-a-2b-2} \left[2^a - 2(2^a-1) + \frac{2^{-a+1}(2^a-1)(2^{a+1}-1)}{3} \right] \\ &= 2^{-a-2b-2} \left[2^a - 2 \cdot 2^a + 2 + 4 \cdot \frac{2^a}{3} - 2 + 2 \cdot \frac{2^{-a}}{3} \right] \\ &= 2^{-a} 2^{-2b} 2^{-2} \left[2^a \left(1 - 2 + \frac{4}{3} \right) + 2^{-a} \left(\frac{2}{3} \right) \right] \\ &= 2^{-2b} 2^{-2} \left[\left(\frac{1}{3} \right) + 2^{-2a} \left(\frac{2}{3} \right) \right] \\ &= \left(\frac{2^{-2b}}{12} \right) (1 + 2^{-2a+1}) \end{aligned} \quad (14-113)$$

The variance of e_r is thus

$$\begin{aligned} \sigma_{e_r}^2 &= E[e_r^2] - E^2[e_r] \\ &= \left(\frac{2^{-2b}}{12} \right) (1 + 2^{-2a+1}) - \left(\frac{2^{-b-a}}{2} \right)^2 \\ &= \left(\frac{2^{-2b}}{12} \right) (1 + 2 \cdot 2^{-2a} - 3 \cdot 2^{-2a}) \\ &= \left(\frac{2^{-2b}}{12} \right) (1 - 2^{-2a}) \end{aligned} \quad (14-114)$$

Since in all practical cases, $a > 4$, then

$$2^{-2a} \ll 1 \quad (14-115)$$

and hence

$$\sigma_{e_r}^2 \doteq \frac{2^{-2b}}{12} \quad (14-116)$$

which is the same result for rounding of continuous signals. In the continuous case the expected value of the round-off error is zero. In a typical case, $b = 8$ and $a = 8$, then

$$\begin{aligned} E[e_r] &= \frac{2^{-b-a}}{2} = 2^{-17} \\ &\doteq 7.62939 \times 10^{-6} \end{aligned} \quad (14-117)$$

and

$$\begin{aligned}\sigma_{e_r}^2 &= \frac{2^{-2b}}{12} = \frac{2^{-16}}{12} \\ &= 1.27157 \times 10^{-6} \\ \sigma_{e_r} &= 1.12764 \times 10^{-3} \gg E[e_r]\end{aligned}\quad (14-118)$$

Consequently, we approximate the round-off of products by using the continuous results derived earlier.

Output Quantization Noise

In the preceding two sections we examined the digital filter input and product-term quantization effects. These quantization errors were modeled as additive input signals in Figures 14-12e and 14-14c. In this section we analyze the effect of these error sources on the output of the digital filter.

Consider the digital filter of Figure 14-16a. This model for a digital filter assumes Q summing junctions, each modeled after Figure 14-14d. In the model $e_0(n)$ represents the filter input quantizer. If we represent the transfer function from the filter input to the output of the i th summing junction by $F_i(z)$, and the transfer function from the output of the i th summing junction to the filter output by $G_i(z)$, we may determine the effect of each individual product term error as shown in Figure 14-16b. Using standard z -transform notation, we have

$$Y^q(z) = X(z)D(z) + E_o(z)D(z) + \sum_{i=1}^Q E_i(z)G_i(z) \quad (14-119)$$

where $E_i(z) = z[e_i(n)]$. However, if we use the equivalent noise model of Figure 14-16c,

$$Y^q(z) = X(z)D(z) + E_n(z) \quad (14-120)$$

Hence the output noise is

$$E_n(z) = E_o(z)D(z) + \sum_{i=1}^Q E_i(z)G_i(z) \quad (14-121)$$

and is depicted in the quantization noise model of Figure 14-16d. Hence we may write

$$E_n(z) = \sum_{i=0}^Q E_{ni}(z) \quad (14-122)$$

where

$$\begin{aligned}E_{ni}(z) &= E_i(z)G_i(z), \quad i = 0, Q \\ G_0(z) &= D(z)\end{aligned}$$

We may examine $E_{ni}(z)$ in several ways. First let us write in the time domain

$$e_{ni}(n) = \sum_{j=0}^{\infty} g_i(j)e_i(n-j) \quad (14-123)$$

Hence

$$\begin{aligned} |e_{ni}(n)| &\leq \sum_{j=0}^{\infty} |g_i(j)||e_i(n-j)| \\ &\leq |e_i(n)|_{\max} \sum_{j=0}^{\infty} |g_i(j)| \end{aligned} \quad (14-124)$$

But we saw in an earlier section that if each product is rounded separately (see Figure 14-14c), then

$$\begin{aligned} |e_i(n)|_{\max} &= L_i |e_r|_{\max} \\ &= L_i \cdot \frac{2^{-b}}{2} \end{aligned} \quad (14-125)$$

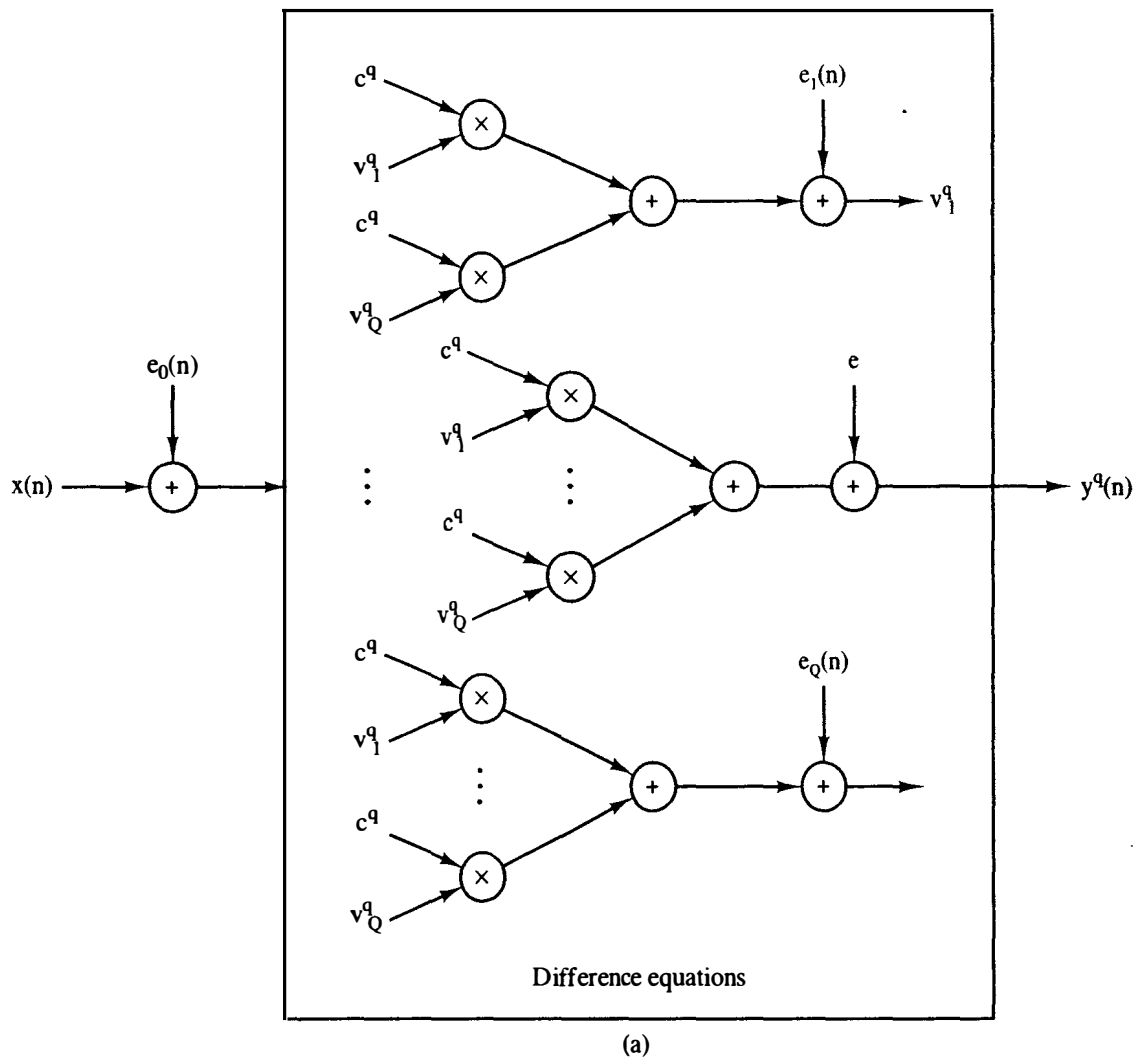


Figure 14-16 Filter noise models.

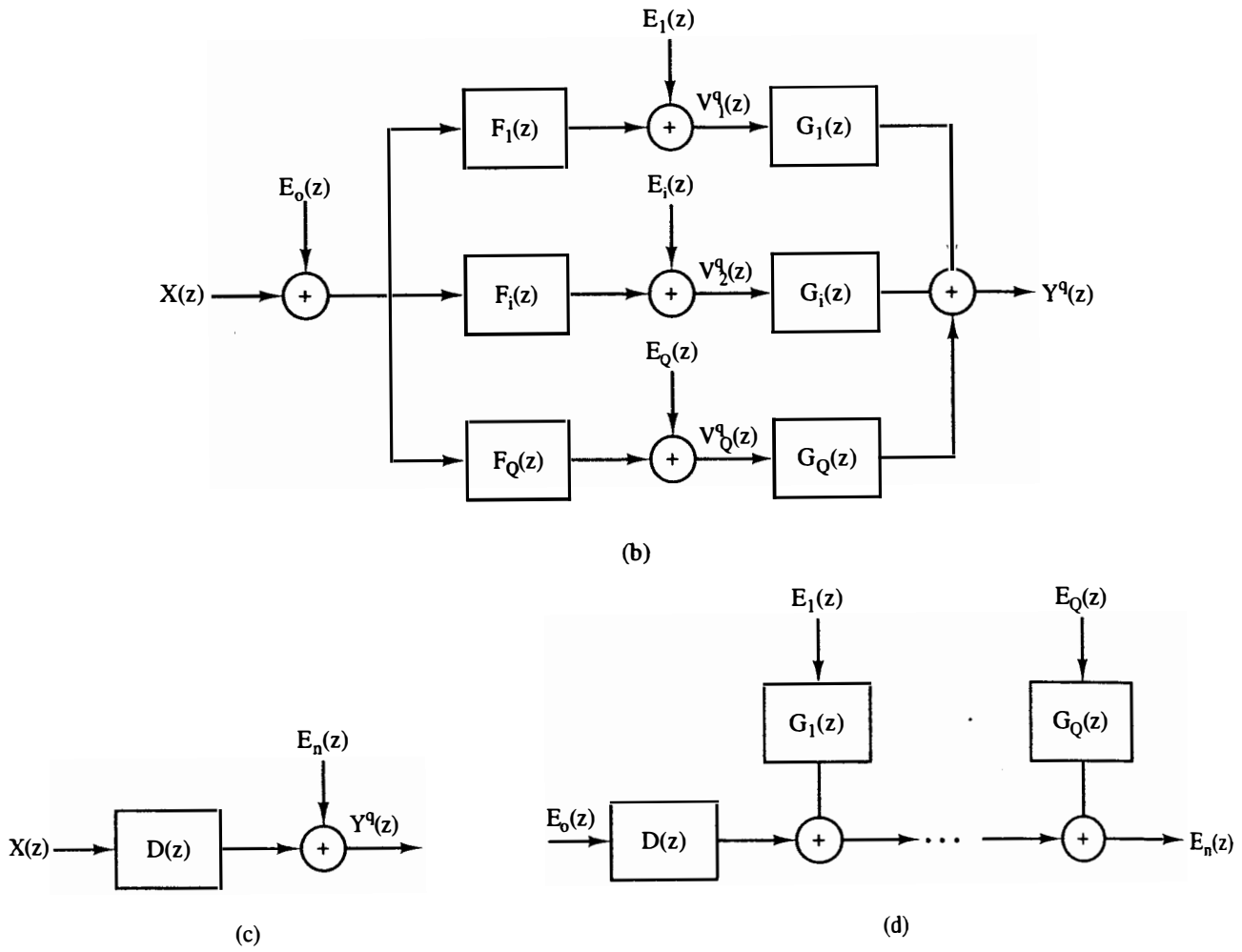


Figure 14-16 (continued)

Consequently,

$$|e_{ni}(n)| \leq \left(2^{-b} \cdot \frac{L_i}{2}\right) \sum_{j=0}^{\infty} |g_i(j)| \quad (14-126)$$

and

$$\begin{aligned} e_n(n) &= \sum_{i=0}^Q e_{ni}(n) \\ |e_n(n)| &\leq \sum_{i=0}^Q |e_{ni}(n)| \\ &\leq \sum_{i=0}^Q \left(2^{-b} \cdot \frac{L_i}{2}\right) \sum_{j=0}^{\infty} |g_i(j)| \end{aligned}$$

So that

$$|e_n(n)| \leq \frac{2^{-b}}{2} \sum_{i=0}^Q \left(L_i \sum_{j=0}^{\infty} |q_i(j)| \right) \quad (14-127)$$

is an *absolute upper bound* on the magnitude of the round-off noise at the output of the digital filter. This is a very pessimistic bound because it assumes, at each sampling instance that the round-off noise at every quantizer is its maximum value and its worst-case sign, so that it drives the filter output to its *worst-case* value.

We may relax the condition of the worst-case *sign*, leaving the worst-case magnitude at each round-off quantizer, and we essentially apply a step input of magnitude $2^{-b}/2$ at each quantizer. Since

$$e_{ni}(n) = \sum_{j=0}^{\infty} g_i(j) e_i(n-j) \quad (14-128)$$

and

$$\begin{aligned} e_i(n-j) &= L_i \cdot \frac{2^{-b}}{2}, & n-j \geq 0 \\ e_i(n-j) &= 0, & n-j < 0 \end{aligned}$$

or

$$e_{ni}(n) = \left(2^{-b} \cdot \frac{L_i}{2}\right) \sum_{j=0}^n g_i(j)$$

and for large n ,

$$|e_{ni}(\infty)| = \left(2^{-b} \cdot \frac{L_i}{2}\right) \left| \sum_{j=0}^{\infty} g_i(j) \right| \quad (14-129)$$

But

$$\sum_{j=0}^{\infty} g_i(j) = G(z)|_{z=1} = G(1)$$

Consequently,

$$|e_n(n)| \leq \frac{2^{-b}}{2} \sum_{i=0}^Q L_i |G_i(1)| \quad (14-130)$$

which is a bit more realistic than our absolute upper bound. We call this the *steady-state* bound.

An alternative derivation of this equation results after applying a step input error of magnitude $2^{-b}/2$ and then determining the steady-state error equation:

$$E_i(z) = \frac{L_i \cdot 2^{-b}/2}{1 - z^{-1}} \quad (14-131)$$

then

$$E_{ni}(z) = \frac{L_i \cdot 2^{-b}/2}{1 - z^{-1}} G_i(z) \quad (14-132)$$

But by the final-value theorem,

$$\begin{aligned} e_{ni}(\infty) &= \lim_{z \rightarrow 1} (1 - z^{-1}) E_{ni}(z) \\ &= L_i \cdot \frac{2^{-b}}{2} G_i(z) \Big|_{z=1} \\ &= \left(L_i \cdot \frac{2^{-b}}{2} \right) G_i(1) \end{aligned} \quad (14-133)$$

So

$$\epsilon_n(\infty) = \frac{2^{-b}}{2} \sum_{i=0}^Q L_i G_i(1)$$

Hence

$$|\epsilon_n(\infty)| \leq \frac{2^{-b}}{2} \sum_{i=0}^Q L_i |G_i(1)| \quad (14-134)$$

Earlier in this chapter we examined the quantization characteristics of several number systems. There we noted that the quantization error for the round-off case could be modeled as a uniformly distributed random noise with zero mean value and variance

$$\sigma_{e_r}^2 = \frac{2^{-2b}}{12} \quad (14-135)$$

Consequently, here we abandon our deterministic approach used in our derivation of absolute upper and steady-state bounds and use statistical methods to analyze the output round-off error.

First, let us define some terminology. The autocovariance of a number sequence $\{x(n)\}$ is

$$Q_x(k - l) = E[x(k)x(l)] \quad (14-136)$$

Note that $Q_x(0) = E[x^2(n)]$. The spectral density can thus be defined as the z-transform of $\{Q(n)\}$

$$S_x(z) = \sum_{n=-\infty}^{\infty} Q_x(n) z^{-n} \quad (14-137)$$

so that

$$Q_x(n) = \frac{1}{2\pi j} \oint S_x(z) z^{n-1} dz \quad (14-138)$$

If we substitute $z = e^{j\omega T}$, then $dz = jTe^{j\omega T} d\omega$ and

$$Q_x(0) = \frac{1}{\omega_s} \int_0^{2\pi} S_x(e^{j\omega T}) d\omega \quad (14-139)$$

Hence

$$E[x^2(n)] = \frac{1}{\omega_s} \int_0^{2\pi} S_x(e^{j\omega T}) d\omega \quad (14-140)$$

It is well known [3] that a filter with transfer function $D(z)$ will have an output spectral density

$$S_y(z) = S_x(z)D(z)D\left(\frac{1}{z}\right) \quad (14-141)$$

Consequently,

$$\begin{aligned} E[y^2(n)] &= \frac{1}{\omega_s} \int_0^{2\pi} S_y(e^{j\omega T}) d\omega \\ &= \frac{1}{\omega_s} \int_0^{2\pi} S_x(e^{j\omega T}) D(e^{j\omega T}) D(e^{-j\omega T}) d\omega \\ &= \frac{1}{\omega_s} \int_0^{2\pi} S_x(e^{j\omega T}) |D(e^{j\omega T})|^2 d\omega \\ &= \sigma_y^2 \quad \text{if } E[y(n)] = 0 \end{aligned} \quad (14-142)$$

For round-off noise analysis

$$\begin{aligned} Q_{e_r}(0) &= E[e_r^2] = \frac{2^{-2b}}{12} \\ Q_{e_r}(n) &= 0, \quad n \neq 0 \end{aligned} \quad (14-143)$$

then

$$S_{e_r}(z) = \sum_{n=-\infty}^{\infty} Q_{e_r}(n) z^{-n} = \frac{2^{-2b}}{12} \quad (14-144)$$

If e_r is the input round-off error, then at the output of the filter

$$\begin{aligned} E[e_n^2(n)] &= \frac{1}{\omega_s} \int_0^{2\pi} S_{e_r}(e^{j\omega T}) |D(e^{j\omega T})|^2 d\omega \\ &= \frac{2^{-2b}}{12\omega_s} \int_0^{2\pi} |D(e^{j\omega T})|^2 d\omega \end{aligned} \quad (14-145)$$

But

$$\begin{aligned} \frac{1}{\omega_s} \int_0^{2\pi} |D(e^{j\omega T})|^2 d\omega &= \frac{1}{2\pi j} \oint D(z) D\left(\frac{1}{z}\right) \frac{dz}{z} \\ &= \sum_{m=0}^{\infty} d^2(m) \end{aligned}$$

Consequently,

$$\sigma_{e_n}^2 = E[e_n^2(n)] = \frac{2^{-2b}}{12} \sum_{m=0}^{\infty} d^2(m) \quad (14-146)$$

If we apply these results to the model of Figure 14-16,

$$\sigma_{e_{ni}}^2 = \sigma_{e_i}^2 \sum_{m=0}^{\infty} g_i^2(m) \quad (14-147)$$

But since the input random noise sources e_i have zero mean, then

$$\begin{aligned} \sigma_{e_n}^2 &= \sum_{i=0}^Q \sigma_{e_{ni}}^2 = \sum_{i=0}^Q \sigma_{e_i}^2 \sum_{m=0}^{\infty} g_i^2(m) \\ &= \frac{2^{-2b}}{12} \sum_{i=0}^Q \sum_{m=0}^{\infty} g_i^2(m) \\ &= \frac{2^{-2b}}{12} \sum_{i=0}^Q \frac{1}{\omega_s} \int_0^{2\pi} |G_i(e^{j\omega T})|^2 d\omega \\ &= \frac{2^{-2b}}{12} \sum_{i=0}^Q \frac{1}{\pi j 2} \oint G_i(z) G_i\left(\frac{1}{z}\right) \frac{dz}{z} \end{aligned} \quad (14-148)$$

These relationships may be evaluated by a computing algorithm described in Ref. 4.

An important measure of a digital filter's performance is its signal-to-noise ratio. If we compute the ratio of the variance of the output signal to the output noise,

$$\begin{aligned} \frac{\sigma_y^2}{\sigma_{e_n}^2} &= \frac{(1/\omega_s) \int_0^{2\pi} S_x(e^{j\omega T}) |G_0(e^{j\omega T})|^2 d\omega}{(2^{-2b}/12) \sum_{i=0}^Q (1/\omega_s) \int_0^{2\pi} |G_i(e^{j\omega T})|^2 d\omega} \\ &= 12 \cdot 2^{2b} \frac{\int_0^{2\pi} S_x(e^{j\omega T}) |G_0(e^{j\omega T})|^2 d\omega}{\sum_{i=0}^Q \int_0^{2\pi} |G_i(e^{j\omega T})|^2 d\omega} \end{aligned} \quad (14-149)$$

Note that the ratio is dependent upon the specific input signal. If we choose a random white noise input bounded by ± 1.0 , then $S_x(e^{j\omega T}) = 1$ and

$$\frac{\sigma_y^2}{\sigma_{e_n}^2} = \frac{12 \cdot 2^{2b}}{1 + \sum_{i=1}^Q \frac{\int_0^{2\pi} |G_i(e^{j\omega T})|^2 d\omega}{\int_0^{2\pi} |G_0(e^{j\omega T})|^2 d\omega}} \quad (14-150)$$

14.5 LIMIT CYCLES

Definitions

A *limit cycle* is a condition of sustained oscillation in a closed-loop system caused by nonlinearities within the loop. Consider the first-order digital filter of Figure 14-17. Let us examine the behavior of the filter if $b = 3$:

$$m(n) = (s.m_1 \ m_2 \ m_3)_{2\text{cns}} \quad (14-151)$$

and

$$b_1 = -0.6$$

Assume that at time zero ($n = 0$) an input pulse of value $x(n) = (0.100)_{2\text{cns}} = (0.5)_{10}$ is applied, and subsequent input values are zero. The filter signal $m(n)$ ideally would decay to zero and force the output $y(n)$ to zero as well. However, with the round-off quantizer in the loop:

n	$m(n-1)$	$-b_1 m(n-1)$	$Q_r^3(-b_1 m(n-1))$	$m(n)$
0	0	0	0	4/8
1	4/8	2.4/8	2/8	2/8
2	2/8	1.2/8	1/8	1/8
3	1/8	0.6/8	1/8	1/8
4	1/8	0.6/8	1/8	1/8
5	etc.	etc.	etc.	etc.

Consequently, the signal $m(n)$ never reaches zero as in the ideal case. A truncation quantizer *would* have produced a zero value for $m(3)$. A digital filter that performs

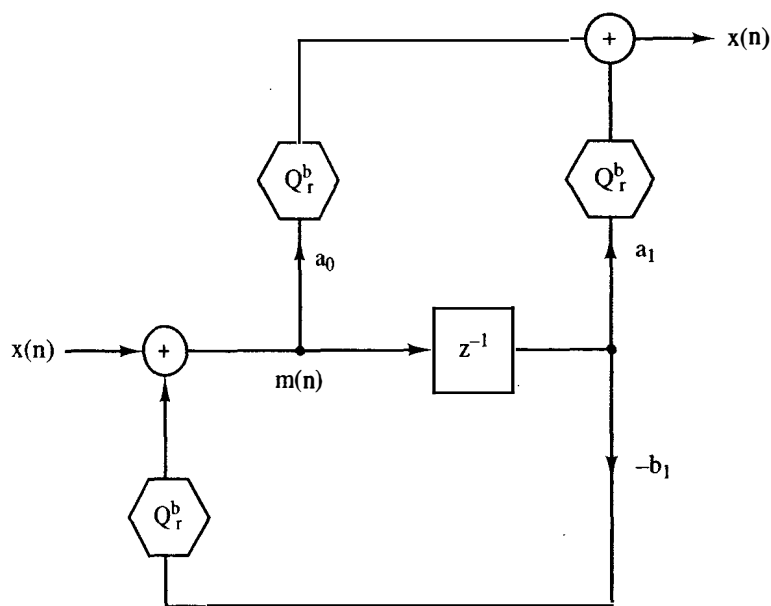


Figure 14-17 First-order 1D filter.

as shown above is said to possess a *deadband*. The deadband effectively *changes* the value of the feedback coefficient; in this case to $b_1 = -1.0$. We know that

$$\frac{M(z)}{X(z)} = \frac{1}{1 + b_1 z^{-1}} \quad (14-152)$$

and

$$m(n) = x(n) - b_1 m(n-1) \quad (14-153)$$

The impulse response is thus

$$\begin{aligned} m(n) &= \cancel{x(n)} - b_1(\cancel{x(n-1)} - b_1 m(n-2)) \\ &= (b_1)^n \end{aligned} \quad (14-154)$$

in the ideal case. In the case of the round-off quantizer,

$$m(n) = Q_r^b(-b_1 m(n-1)) \quad (14-155)$$

But we know that, for round-off,

$$|-b_1 m(n-1)| - |Q_r^b(-b_1 m(n-1))| \leq 2^{-b-1}$$

Consequently,

$$|-b_1 m(n-1)| - |m(n)| \leq 2^{-b-1}$$

If $b_1 > 0$, then its effective value in the deadband will be 1.0 and $m(n) = -m(n-1)$; if $b_1 < 0$, -1.0 and $m(n) = m(n-1)$. Hence

$$|m(n)| \leq \frac{2^{-b-1}}{1 - |b_1|} \quad (14-156)$$

Remember that $m(n)$ is an integer ($\times 2^{-b}$) and hence a deadband does not exist if $|b_1| < 0.5$.

Now let us consider the filter structure of Figure 14-18 under large-signal conditions. Suppose that we consider a 3-bit two's-complement number system with

$$b_1 = +1.5$$

$$b_2 = +0.5$$

and let an input sequence of $+5/8, 0, 0, 0, \dots$ be applied to the network. The poles of the filter are

$$(z^2 + 1.5z + 0.5) = (z + 1.0)(z + 0.5) \quad (14-157)$$

so that the ideal filter impulse response is stable:

n	$m(n-1)$	$Q_r^3(-b_1 m(n-1))$	$m(n-2)$	$Q_r^3(-b_2 m(n-2))$	$m(n)$
0	0	0	0	0	5/8
1	5/8	-7.5/8	0	0	-7.5/8
2	-7.5/8	11.25/8	5/8	-2.5/8	8.75/8
3	8.75/8	-13.125/8	-7.5/8	3.75/8	-9.375/8
4	-9.375/8	14.0625/8	8.75/8	-4.375/8	9.6875/8
5	9.6875/8	-14.53125/8	-9.375/8	4.6875/8	-9.84375/8
6	-9.84375/8	14.765625/8	9.6875/8	-4.84375/8	9.921765/8
7	9.921875/8	-14.8828125/8	-9.84375/8	4.921875/8	-9.9609375/8
8	-9.9609375/8	14.941406/8	9.921875/8	-4.9608375/8	9.9805688/8

Now suppose that we apply the round-off quantizers to the least significant bits (see Figure 14-5b) and the overflow characteristic to the most significant bits of the adder output (see Figure 14-8). The following sequence is produced:

n	$m(n-1)$	$Q_r^3(-b_1 m(n-1))$	$m(n-2)$	$Q_r^3(-b_2 m(n-2))$	$m(n)$
0	0	0	0	0	5/8
1	5/8	-8/8	0	0	-8/8
2	-8/8	12/8	5/8	-3/8	9/8 → -7/8
3	-7/8	11/8	-8/8	4/8	15/8 → -1/8
4	-1/8	2/8	-7/8	4/8	6/8
5	6/8	-9/8	-1/8	1/8	-8/8
6	-8/8	12/8	6/8	-3/8	9/8 → -7/8
7	-7/8	11/8	-8/8	4/8	15/8 → -1/8

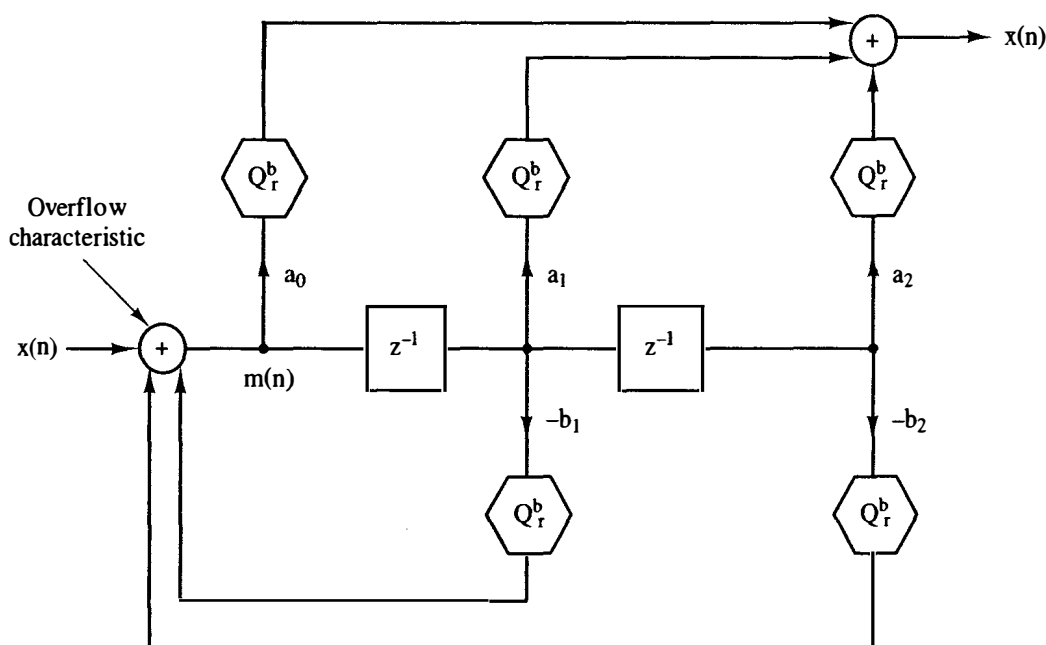


Figure 14-18 Second-order 1D filter with one quantizer per multiplier.

Note that line $n = 7$ is identical to line $n = 3$. Hence a large limit cycle in the range

$$-1 \leq m(n) \leq +0.75 \quad (14-158)$$

has been produced, essentially by the overflow characteristic of the two's-complement number system adder as shown in the last column. For example, $n = 2$:

$$\begin{array}{rcl} Q_r^3(-b_1 m(n-1)) & = & 12/8 = 1.100 \\ + Q_r^3(-b_2 m(n-2)) & = & -3/8 = 1.101 \\ \hline m(n) & & 9/8 \quad \underbrace{11.001}_{m(n)} \end{array} \quad (14-159)$$

Hence

$$\begin{aligned} m(n) &= (1.001)_{2\text{cns}} = -(0.111)_2 \\ &= -(7/8)_{10} \end{aligned} \quad (14-160)$$

Note that the quantizers Q_r^b in Figure 14-18 affect only the least significant bits, while the adder performs the overflow characteristic. The limit cycles produced by the overflow characteristic are called *overflow oscillations*.

Classification of Quantization Errors

The two cases shown above were examples of zero-input limit cycles. Consider the classification scheme of Table 14-1. The table illustrates the effect of signal amplitude quantizers and overflow characteristics on the output of a digital filter. The term *limit cycle* is commonly used to mean the small-signal limit cycles seen around the 2^{-b} signal level as described in the first-order example above. The term *quantization noise* refers to the error type described in Section 14.4. The term *overflow noise* has been included to mean cases in which an occasional overflow will add a large noise "spike" into the filter output signal. In the remainder of this section we concentrate on analyzing the limit cycle and overflow oscillation phenomena.

TABLE 14-1 CLASSIFICATION OF QUANTIZATION ERRORS

Input condition	Nonlinearity type	
	Quantizer	Overflow
Zero input	Limit cycles	Overflow oscillations
Deterministic input		Overflow noise
Periodic	Limit cycles	
Nonperiodic	Quantization noise	
Stochastic input	Quantization noise	Overflow noise

Limit Cycles

Let us now examine a 3D filter of second order as shown in Figure 14-19. Note that double-precision product terms are added together and then quantized to form the output variable $y^q(k)$:

$$y(k) = a_0 Q_r^b(x(k)) + a_1 Q_r^b(x(k-1)) + a_2 Q_r^b(x(k-2)) - b_1 Q_r^b(y(k-1)) - b_2 Q_r^b(y(k-2)) \quad (14-161)$$

For the zero-input limit cycle case

$$y(k) = -b_1 Q_r^b(y(k-1)) - b_2 Q_r^b(y(k-2))$$

And since $y^q(k) = Q_r^b(y(k))$,

$$|Q_r^b(y(k)) - y(k)| \leq 2^{-b-1} \quad (14-162)$$

Then

$$|Q_r^b(y(k)) + b_1 Q_r^b(y(k-1)) + b_2 Q_r^b(y(k-2))| \leq 2^{-b-1} \quad (14-163)$$

Case 1. Suppose that a constant nonzero output level is attained. Then

$$Q_r^b(y(k)) = Q_r^b(y(k-1)) = Q_r^b(y(k-2)) \quad (14-164)$$

and the deadband is

$$|Q_r^b(y(k))| \leq \frac{2^{-b-1}}{|1 + b_1 + b_2|} \quad (14-165)$$

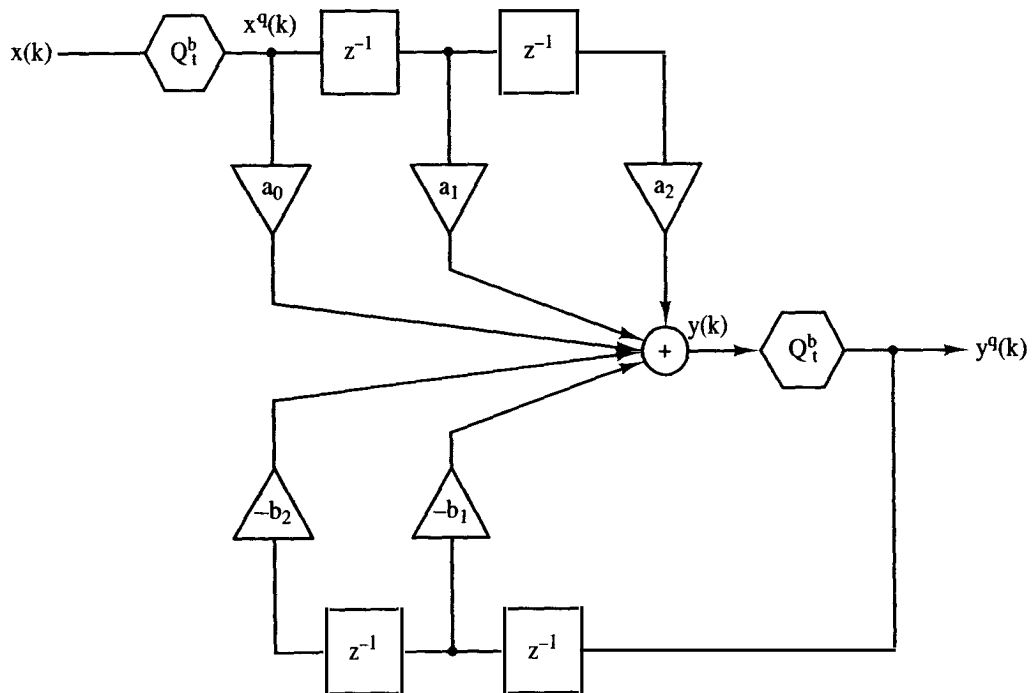


Figure 14-19 3D second-order filter.

Case 2. Suppose that a square-wave limit cycle is attained. Then

$$Q_r^b(y(k)) = -Q_r^b(y(k-1)) = Q_r^b(y(k-2)) \quad (14-166)$$

Consequently,

$$|Q_r^b(y(k))| \times |1 - b_1 + b_2| \leq 2^{-b-1}$$

and

$$|Q_r^b(y(k))| \leq \frac{2^{-b-1}}{|1 - b_1 + b_2|} \quad (14-167)$$

Case 3. Suppose that a sinusoidal limit cycle output is attained. Then the effective value of b_2 will be 1. That is,

$$Q_r^b \cdot b_2 \times Q_r^b(y(k-2)) = Q_r^b \cdot 1 \times Q_r^b(y(k-2)) \quad (14-168)$$

Hence

$$|Q_r^b(y(k-2))| - |b_2 Q_r^b(y(k-2))| \leq 2^{-b-1}$$

and

$$|Q_r^b(y(k-2))| \leq \frac{2^{-b-1}}{1 - |b_2|}$$

Consequently,

$$|Q_r^b(y(k))| \leq \frac{2^{-b-1}}{1 - |b_2|} \quad (14-169)$$

In the cases presented above, we assumed a limit cycle waveform and solved for the magnitude bound of the filter output. Another way to view limit cycles is to consider the digital filter to be a finite-state, synchronous sequential circuit. The state of the circuit S_i is determined by the value of the variables in the delay elements (implemented by flip-flops or RAM cells). In Figure 14-19 there are four delay elements of $b + 1$ bits each so that the number of distinct states for the sequential circuit is

$$N_s = 2^{4(b+1)} \quad (14-170)$$

If $b = 15$ as in Chapter 11, then

$$N_s = 2^{64}$$

a very large number. For the zero-input case the subset of finite states is

$$N_s = 2^{32}$$

which is still over 1 billion.

In Table 14-1 we noted that limit cycles were always small-signal variations in the output signal. Practically speaking, only 3 or 4 bits are usually involved, so that

$$N_s \doteq 2^8 \quad (14-171)$$

which is a manageable number for analysis purposes.

Now examine Figure 14-20. The limit cycle $S_i \rightarrow S_j \rightarrow S_k \rightarrow \dots \rightarrow S_i$ can have an even or odd number of states, M . In case 1 above we assumed $M = 1$; case 2, $M = 2$. The limit cycle can be initiated by a starting state within the cycle (any of the limit cycle states) or from without the cycle as shown in state S_m in the model. Many researchers have sought to describe the behavior of the limit cycles modeled above [5-12]. In what follows we describe some of their results. The reader is referred to the literature for more advanced treatment of the limit cycle problem.

The limit cycle problem is essentially a feedback problem as modeled in Figure 14-21. The direct-form structures (Figure 14-21a-d) may all be described in the zero-input limit cycle case by Figure 14-21e.

$$m^q(k) = Q_2 \cdot Q_1(-b_1 m^q(k-1)) + Q_1(-b_2 m^q(k-2)) \quad (14-172)$$

The quantizers Q_1 or Q_2 are chosen for implementation. That is, if product terms are rounded after multiplication and before terms are added, Q_1 is active and Q_2 is absent. If, however, double-precision products are first added and the final result rounded, Q_2 is active and Q_1 is absent.

Limit cycle bounds. Long and Trick [9] have derived several bounds for limit cycle errors. Consider the 3D direct structure

$$y(k) = \sum_{i=0}^n a_i x(k-i) - \sum_{i=1}^n b_i y(k-i) \quad (14-173)$$

For the zero-input case

$$y(k) = \sum_{i=1}^n b_i y(k-i) \quad (14-174)$$

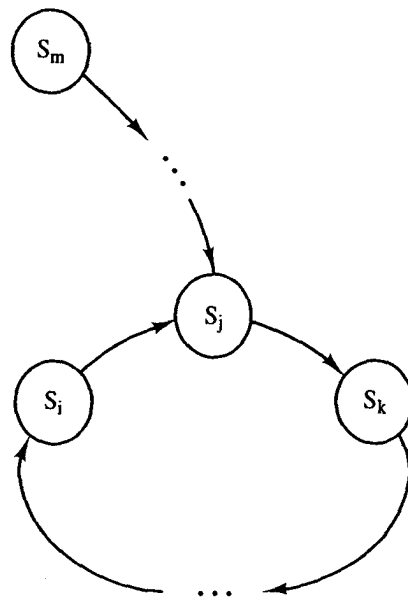


Figure 14-20 Limit cycle model.

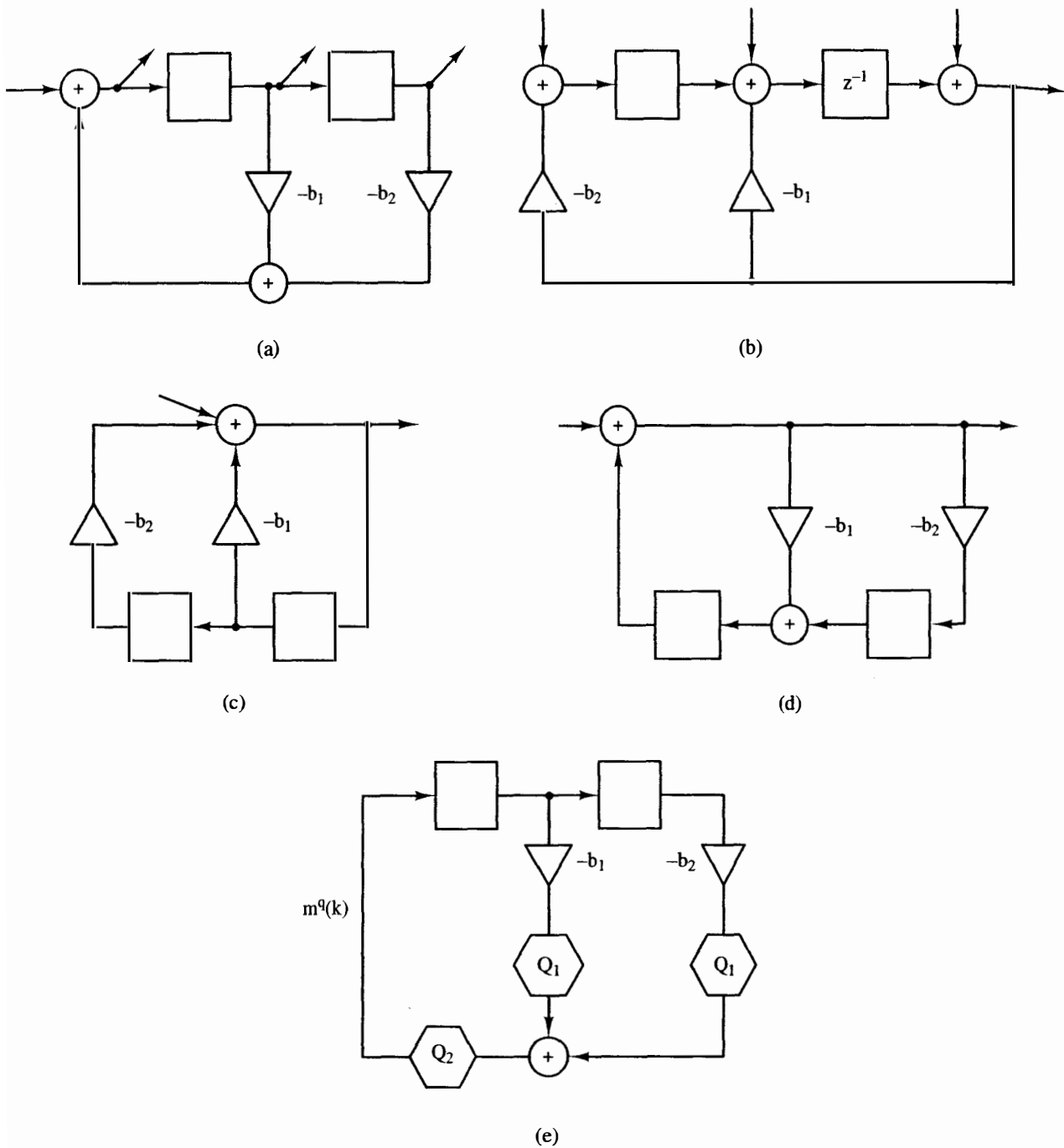


Figure 14-21 Quantization options: (a) 1D; (b) 2D; (c) 3D; (d) 4D; (e) general model for direct forms.

Now adding quantizers at position Q_1 (see Figure 14-21)

$$y^q(k) = - \sum_{i=1}^n Q_r^b(b_i y^q(k-i)) \quad (14-175)$$

But if we define [see (14-1)]

$$Q_r^b(b_i y^q(k-i)) = b_i y^q(k-i) + e_i(k) \quad (14-176)$$

then

$$y^q(k) = - \sum_{i=1}^n (b_i y^q(k-i) + e_i(k)) \quad (14-177)$$

Hence

$$y^q(k) = - \sum_{i=1}^n b_i y^q(k-i) - \sum_{i=1}^n e_i(k) \quad (14-178)$$

So if we consider

$$e(k) = - \sum_{i=1}^n e_i(k) \quad (14-179)$$

to be an input signal into the digital filter, the limit cycle response will be

$$y^q(k) = \sum_{l=-\infty}^k h(k-l)e(l) \quad (14-180)$$

where $h(k)$ is the filter impulse response. We know that the limit cycle is periodic, say M cycles:

$$e(l) = e(l+M) \quad (14-181)$$

Hence

$$y^q(k) = \sum_{j=0}^{\infty} \left[\sum_{l=k-(j+1)M+1}^{k-jM} h(k-l)e(l) \right] \quad (14-182)$$

If $p = k - jM - l$,

$$\begin{aligned} y^q(k) &= \sum_{j=0}^{\infty} \left[\sum_{p=0}^{M-1} h(p+jM)e(k-p) \right] \\ &= \sum_{p=0}^{M-1} e(k-p) \left[\sum_{j=0}^{\infty} h(p+jM) \right] \end{aligned} \quad (14-183)$$

But

$$|e(k-p)| \leq n(2^{-b-1}) \quad (14-184)$$

so

$$y^q(k) = n \cdot 2^{-b-1} \sum_{p=0}^{M-1} \left[\sum_{j=0}^{\infty} h(p+jM) \right] \quad (14-185)$$

An absolute upper bound

$$|y^q(k)| \leq n \cdot 2^{-b-1} \sum_{p=0}^{\infty} |h(p)| \quad (14-186)$$

can be found which is the same bound found earlier for quantization noise in (14-127).

For second-order filters as modeled in Figure 14-21,

$$H(z) = \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}} \quad (14-187)$$

Hence

$$h(p) = \oint \frac{z^2}{z^2 + b_1 z + b_2} z^{p-1} dz \quad (14-188)$$

For distinct poles of the filter

$$|y^q(k)| \leq n \cdot 2^{-b-1} \sum_{p=0}^{M-1} \left| \frac{1}{2\left(\frac{b_1^2}{4} - b_2\right)} \right. \\ \times \left[\frac{\left(\frac{-b_1}{2} + \sqrt{\frac{b_1^2}{4} - b_2}\right)^{p+1}}{1 - \left(\frac{-b_1}{2} + \sqrt{\frac{b_1^2}{4} - b_2}\right)^M} - \frac{\left(\frac{-b_1}{2} - \sqrt{\frac{b_1^2}{4} - b_2}\right)^{p+1}}{1 - \left(\frac{-b_1}{2} - \sqrt{\frac{b_1^2}{4} - b_2}\right)^M} \right] \quad (14-189)$$

and for repeated poles

$$|y^q(k)| \leq n \cdot 2^{-b-1} \sum_{p=0}^{M-1} \left| \left(-\frac{b_1}{2}\right)^p \right. \\ \times \left[\frac{k}{1 - \left(-\frac{b_1}{2}\right)^M} + \frac{1 + \left(-\frac{b_1}{2}\right)^M (M-1)}{1 - \left(-\frac{b_1}{2}\right)^{M^2}} \right] \quad (14-190)$$

Hence the absolute bounds for limit cycles for second-order filters become:

Case 1. If $b_2 \leq 0$, or if $b_2 > 0$ and $2\sqrt{b_2} \leq |b_1|$,

$$|y^q(k)| \leq \frac{2^{-b}}{1 - |b_1| + b_2} \quad (14-191)$$

Case 2. If $b_2 > 0$ and $2b_2\sqrt{(2/\sqrt{b_2}) - 1} \leq |b_1| \leq 2\sqrt{b_2}$,

$$|y^q(k)| \leq \frac{2^{-b}}{(1 - \sqrt{b_2})^2} \quad (14-192)$$

Case 3. If $b_2 > 0$ and $|b_1| \leq 2b_2\sqrt{(2/\sqrt{b_2}) - 1}$,

$$|y^q(k)| \leq \frac{(1 + \sqrt{b_2})2^{-b}}{(1 - b_2)\sqrt{1 - b_1^2/4b_2}} \quad (14-193)$$

Absence of limit cycles [10]. Suppose that we model a digital filter with one quantizer as shown in Figure 14-22. Now

$$X(z) = W(z)Y(z) \quad (14-194)$$

If

$$Q(0) = 0$$

$$0 \leq \frac{Q(z)}{x} \leq k, \quad x \neq 0 \quad (14-195)$$

$$\operatorname{Re} W(z_l) - \frac{1}{k} < 0, \quad l = 0, \quad [M/2]$$

where $[x]$ indicates the integer part of x and

$$z_l = e^{j(2\pi/M)l}$$

then limit cycles of length M are *absent* from the digital filter [10].

If we apply these properties to the second-order digital filter of Figure 14-21e with Q_2 quantizer, then

$$W(z) = b_1 z^{-1} + b_2 z^{-2} \quad (14-196)$$

and

$$\operatorname{Re} W(z) = b_1 \cos \left[\left(\frac{2\pi}{M} \right) l \right] + b_2 \cos \left[\left(\frac{4\pi}{M} \right) l \right] \quad (14-197)$$

If $M = 1$, stationary limit cycles cannot exist if

$$b_1 + b_2 - \frac{1}{k} < 0 \quad (14-198)$$

Limit cycles of length $M = 2$ are absent if

$$-b_1 + b_2 - \frac{1}{k} < 0 \quad (14-199)$$

is also valid. Continuing this process, all limit cycles will be absent if

$$b_1 \cos \phi + b_2 \cos 2\phi - \frac{1}{k} < 0 \quad (14-200)$$

for $0 \leq \phi \leq \pi$.

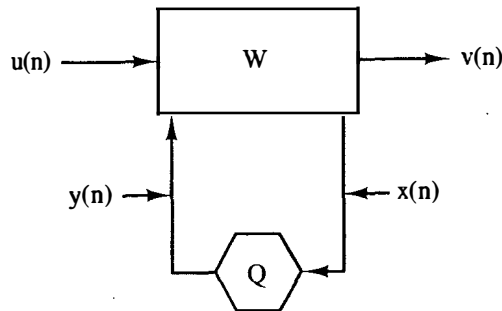


Figure 14-22 Digital filter with one quantizer.

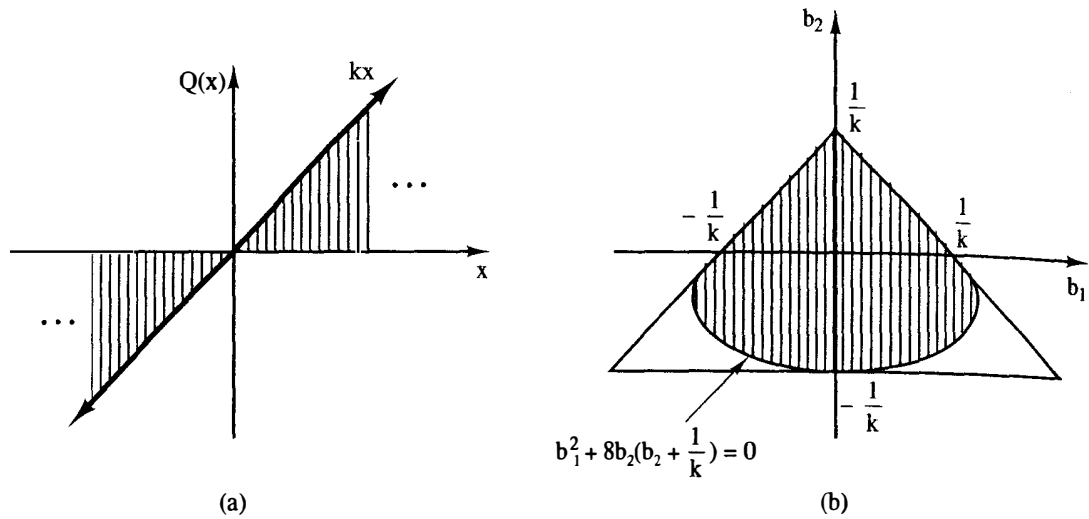


Figure 14-23 Absence of limit cycles: (a) nonlinear characteristic; (b) filter coefficient space.

These results are summarized in Figure 14-23. The quantizer characteristic must fall in the shaded area. If so, the foregoing conditions are shown in the coefficient space. If the coefficients fall within the shaded area, no limit cycles will exist. Note that for the round-off quantizer, $k = 2$; the truncation quantizer, $k = 1$.

Overflow Oscillations

Overflow oscillations must not be allowed to occur in a digital filter. Their avoidance can take three approaches:

1. Scale the input to the filter so that only small signal levels exist in the filter and overflow never occurs. This procedure will be discussed in Section 14.7.
2. Design the adder unit so that its overflow characteristic will not produce oscillations [13]. The overflow characteristic must lie in the shaded area of Figure 14-24. Note that the two's-complement and signed-magnitude adders do *not* satisfy this condition.
3. Find a digital filter structure which is free of overflow oscillations, even when implemented with two's-complement arithmetic. This is the technique we will investigate here.

Conditions for absence of overflow oscillations [14]. Let us represent the digital filter in state-variable notation

$$\mathbf{x}(n+1) = A\mathbf{x}(n) + \mathbf{b}u(n) \quad (14-201)$$

The overflow characteristic may be imposed as follows:

$$\mathbf{x}(n+1) = G(A\mathbf{x}(n) + \mathbf{b}u(n)) \quad (14-202)$$

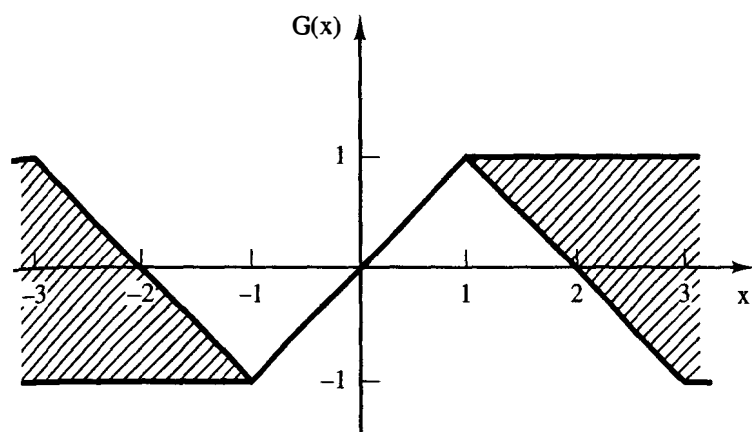


Figure 14-24 Absence of overflow oscillations.

Theorem 1. If there is a diagonal matrix D with positive diagonal elements and $D - A^T D A$ is positive definite, overflow oscillations are impossible.

Theorem 2. Let A be a 2×2 matrix with eigenvalues $|\lambda| < 1$. There exists a positive definite diagonal matrix D for which $D - A^T D A$ is positive definite if and only if

$$a_{12} a_{21} \geq 0 \quad (14-203)$$

or

$$a_{12} a_{21} < 0 \quad \text{and} \quad |a_{11} - a_{22}| + \det(A) < 1 \quad (14-204)$$

Example 14.1

Consider the second-order filter of Figure 14-18.

$$A = \begin{bmatrix} 0 & 1 \\ -b_2 & -b_1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (14-205)$$

$$\mathbf{x}(n) = \begin{bmatrix} m(n-2) \\ m(n-1) \end{bmatrix}$$

For stability it is known that

$$\begin{aligned} 1 - b_2 &> 0 \\ 1 + b_1 + b_2 &> 0 \\ 1 - b_1 + b_2 &> 0 \end{aligned} \quad (14-206)$$

Using Theorem 2, overflow oscillations are absent if

$$a_{12} a_{21} = -b_2 \geq 0 \quad (14-207)$$

or if

$$-b_2 < 0 \quad \text{and} \quad |a_{11} - a_{22}| + \det(A) = b_1 + b_2 < 1 \quad (14-208)$$

Hence overflow oscillations are absent if

$$|b_1| + |b_2| < 1 \quad (14-209)$$

Unfortunately, this condition cannot always be met when designing second-order modules.

Structure with absent overflow oscillations. For any stable filter there is always a two's-complement implementation in which overflow oscillations are absent. The Liapunov stability theory states that the unique solution P to

$$P = A^T P A + I \quad (14-210)$$

is positive definite. If T is a symmetric square root of P^{-1} , then

$$P^{-1} = I - (T^{-1} A T)^T I (T^{-1} A T) \quad (14-211)$$

This T is a coordinate transform that will produce a new A which meets the conditions of Theorem 1. However, this solution will require a large number of multiplications and may not be practical in some applications.

The treatment of the overflow oscillation problem is an important step in the design of a digital filter. The three approaches above may not offer an optimal solution, but their application to a specific design problem can indeed find a practical solution.

14.6 IMPACT OF FINITE WORDLENGTH ON FILTER IMPLEMENTATION

In Chapters 8 through 11, we have discussed the design of digital filter transfer functions:

$$D(z) = \frac{\sum_{i=0}^n a_i z^{-i}}{1 + \sum_{i=1}^n b_i z^{-i}} \quad (14-212)$$

where a_i and b_i are constant, real numbers. In Chapter 12 we displayed several direct and cross-coupled digital filter structures suitable for implementing (14-212). In Chapter 13, we presented techniques for realizing the structures in Chapter 12. These realization methods impose finite-wordlength constraints on (14-212). The nature of these finite-wordlength constraints was examined earlier in this chapter. Here we explore the impact of these finite-wordlength constraints on the filter design and implementation process.

In Chapter 12 it was noted that higher-order filters ($n \geq 4$) are usually implemented as cascaded or paralleled second-order modules in order to avoid the pole-sensitivity problem described in Section 14.3. In avoiding the coefficient-sensitivity problem, we introduce other problems; specifically, pole-zero pairing, module scaling, and module ordering. In what follows we examine each of these new problems and give practical design guidelines for handling them.

14.7 CASCADED SECOND-ORDER MODULES

In implementing (14-212) as cascaded second-order modules, $D(z)$ may be factored into second-order numerator and denominator terms:

$$D(z) = \frac{\prod_{i=1}^m a_i(z)}{\prod_{i=1}^m \beta_i(z)} = \prod_{i=1}^m A_i(z) \quad (14-213)$$

where, from (12-14),

$$\begin{aligned} \alpha_i(z) &= \alpha_{i0} + \alpha_{i1}z^{-1} + \alpha_{i2}z^{-2} \\ \beta_i(z) &= 1 + \alpha_{i3}z^{-1} + \alpha_{i4}z^{-2} \end{aligned} \quad (14-214)$$

and m is the smallest integer greater than $n/2$. Figure 14-25 displays the cascaded realization. In order to form (14-213), second-order numerator and denominator terms must first be paired; then the pairs must be ordered in cascade. Each second-order module may then be implemented by one of the structures of Chapter 12. Consider the four direct structures of Figure 12-2. These structures have been redrawn in Figure 14-26a–d showing the signal amplitude error sources as modeled in Figure 14-16. Please note that the 1D structure has two error sources per module; and the 3D, one. However, the 1D and 4D structures have similar error behavior, as do the 2D and 3D. Composite error models for these structures are shown in Figure 14-26e and f.

Earlier we emphasized that overflow oscillations can produce disastrous results. In practice, *scaling factors* are introduced into the cascade in order to limit signal amplitudes so that overflow does not occur. Let us examine a cascade of 1D (or 4D) second-order modules with scaling coefficients included in the cascade (see Figure 14-27a). We may simplify the notation by factoring a_0 from (14-212):

$$D(z) = a_0 \prod_{i=1}^m A_i(z) \quad (14-215)$$

where

$$A_i(z) = \frac{1 + \alpha_{i1}z^{-1} + \alpha_{i2}z^{-2}}{1 + \alpha_{i3}z^{-1} + \alpha_{i4}z^{-2}}$$

Consequently,

$$\prod_{i=0}^m s_i = a_0 \quad (14-216)$$

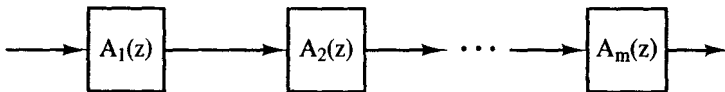
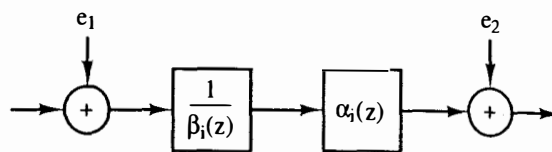
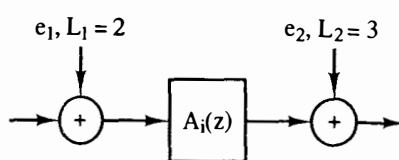
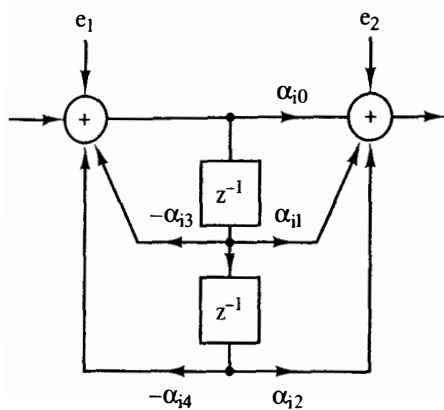
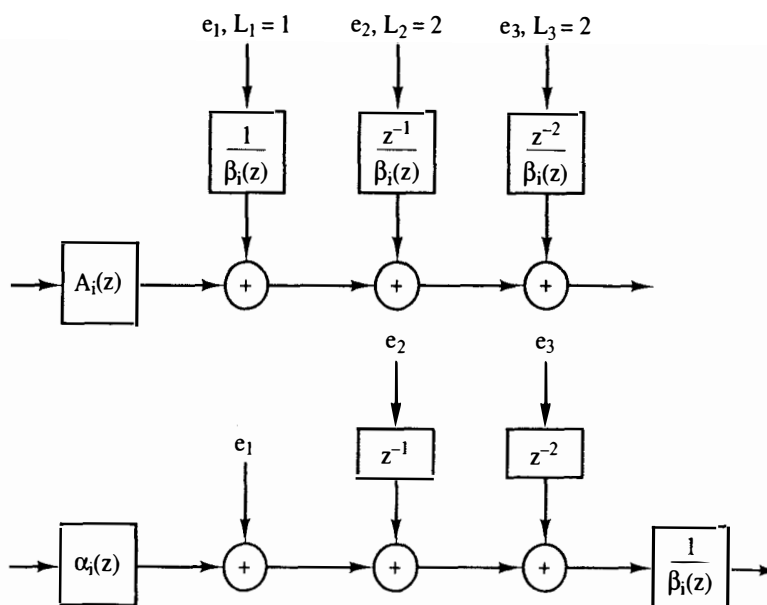
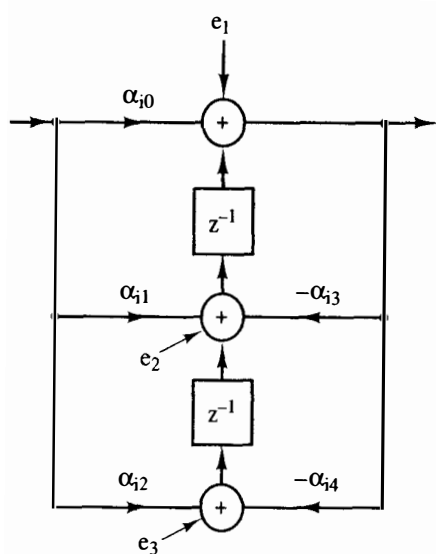


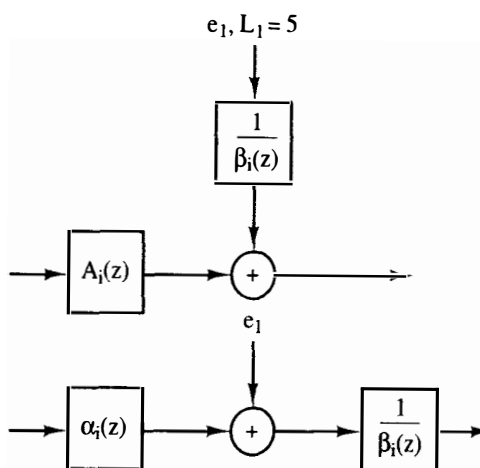
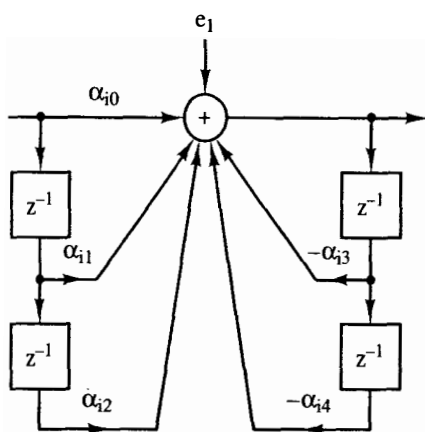
Figure 14-25 Cascaded modules.



(a)



(b)



(c)

Figure 14-26 Direct structure error models: (a) 1D; (b) 2D; (c) 3D; (d) 4D; (e) 1D, 4D composite model; (f) 2D, 3D composite model.

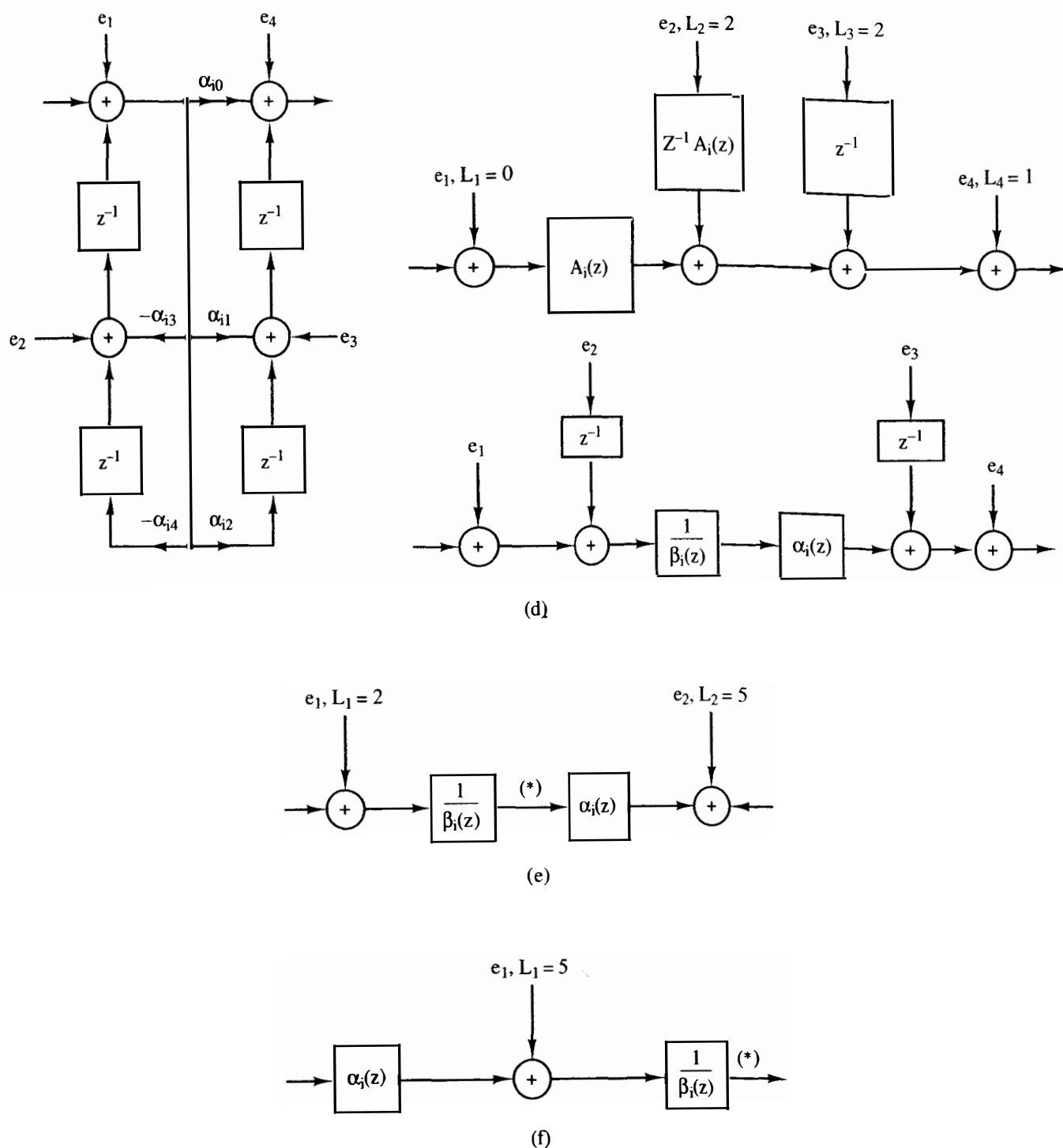


Figure 14-26 (continued)

Previously, we derived several estimates of the output noise of a digital filter as a function of its amplitude error sources. For example, (14-148) states that the variance of the output noise may be computed by

$$\sigma_{e_n}^2 = \frac{2^{-2b}}{12} \sum_{i=0}^Q \frac{1}{2\pi j} \oint G_i(z) G_i\left(\frac{1}{z}\right) \frac{dz}{z} \quad (14-217)$$

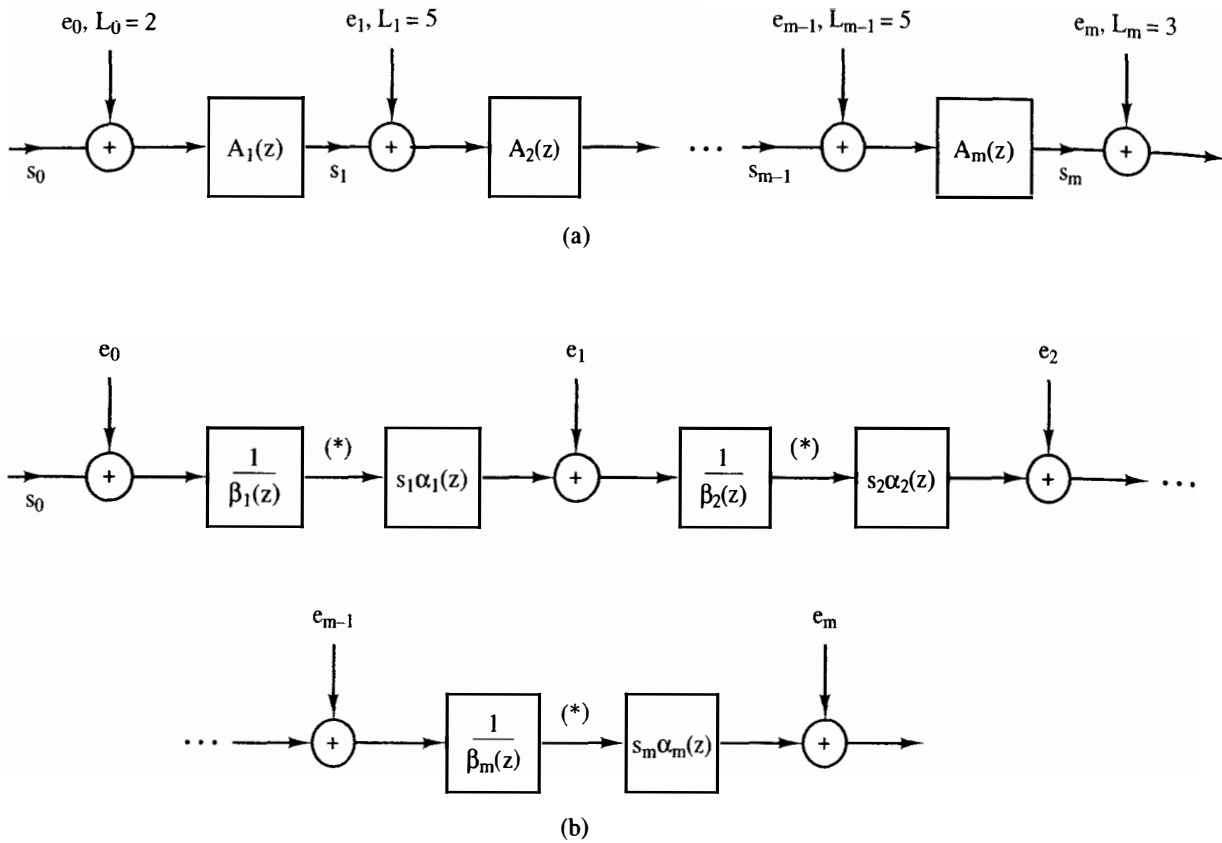


Figure 14-27 1D or 4D cascaded filter: (a) scaling between modules; (b) numerator scaling.

where Q is the number of error sources (round-off quantizers) and $G_i(z)$ is the transfer function from the error source e_i to the filter output. Hence in Figure 14-27b

$$\begin{aligned} \sigma_{e_n}^2 = & \frac{2^{-2b}}{12} \left[L_m + \frac{L_{m-1}}{2\pi j} \oint \frac{s_m \alpha_m(z) s_m \alpha_m(1/z)}{\beta_m(z) \beta_m(1/z)} \frac{dz}{z} \right. \\ & + \frac{L_{m-2}}{2\pi j} \oint \frac{s_m \alpha_m(z) s_{m-1} \alpha_{m-1}(z) s_m \alpha_m(1/z) s_{m-1} \alpha_{m-1}(1/z)}{\beta_m(z) \beta_{m-1}(z) \beta_m(1/z) \beta_{m-1}(1/z)} \frac{dz}{z} \\ & + \dots + \left. \frac{L_0}{2\pi j} \oint \prod_{i=1}^m \frac{s_i \alpha_i(z) s_i \alpha_i(1/z)}{\beta_i(z) \beta_i(1/z)} \frac{dz}{z} \right] \end{aligned} \quad (14-218)$$

or by rearranging terms

$$\sigma_{e_n}^2 = \frac{2^{-2b}}{12} \left[L_m + \sum_{l=1}^m \frac{L_{l-1}}{2\pi j} \oint \prod_{i=l}^m s_i^2 A_i(z) A_i\left(\frac{1}{z}\right) \frac{dz}{z} \right] \quad (14-219)$$

where $L_0 = 2$, $L_m = 3$; otherwise, $L_i = 5$.

The goal of design in a cascade of second-order modules is to minimize (14-219). The parameters may be varied by pairing, scaling, and ordering.

Next let us examine the behavior of a cascade of 2D (or 3D) modules (see Figure 14-28). Using the model in Figure 14-26 and (14-217), we have

$$\begin{aligned} \sigma_{e_n}^2 = & \frac{2^{-2b}}{12} \left[\frac{L_m}{2\pi j} \oint \frac{s_m^2(dz/z)}{\beta_m(z)\beta_m(1/z)} \right. \\ & + \frac{L'_{m-1}}{2\pi j} \oint \frac{s_m^2 s_{m-1}^2 A_m(z) A_m(1/z) dz}{\beta_{m-1}(z)\beta_{m-1}(1/z) z} \\ & + \frac{L_{m-2}}{2\pi j} \oint \frac{s_m^2 s_{m-1}^2 s_{m-2}^2 A_m(z) A_{m-1}(z) A_m(1/z) A_{m-1}(1/z) dz}{\beta_{m-2}(z)\beta_{m-2}(1/z) z} \\ & + \dots + \frac{L_1}{2\pi j} \oint \frac{s_m^2 s_{m-1}^2 \dots s_1^2 A_m(z) \dots A_2(z) A_m(1/z) \dots A_2(1/z) dz}{\beta_1(z)\beta_1(1/z) z} \left. \right] \end{aligned} \quad (14-220)$$

Hence

$$\sigma_{e_n}^2 = \frac{2^{-2b}}{12} \sum_{l=1}^m \frac{L_l}{2\pi j} \oint \frac{\prod_{i=l}^m s_i^2 A_{i+1}(z) A_{i+1}(1/z)}{\beta_l(z)\beta_l(1/z)} \frac{dz}{z} \quad (14-221)$$

where $A_{m+1}(z) = 1$. Again the design goal is to minimize (14-221) by properly pairing, scaling, and ordering the terms of (14-213).

Signal Scaling

In Section 14.5 an example of a digital filter with overflow oscillations was presented. The filter was a 1D structure of second-order employing two's-complement arithmetic. In this example the internal signal $m(n)$ in Figure 14-18 exceeded the dynamic range $[1 > m(n) \geq -1]$ of the fixed-point number system and hence overflow occurred and introduced large-scale oscillations in the structure. The overflow oscillations, in general, may be eliminated by reducing (*scaling*) the input signal to a digital filter. However, if one scales the signals down to very small values, quantization errors become a significant part of the internal signals, and the signal-to-noise ratio is degraded. Consequently, an important design problem is to choose scaling factors that reduce the probability of overflow while maintaining signals at significant levels of the dynamic range.

The insertion of scaling coefficients between cascaded modules has been illustrated in Figures 14-27 and 14-28. These scaling coefficients are chosen such that the magnitude of the internal signal values $[V_i^q(z)]$ of Figure 14-16] does not exceed 1, the overflow limit. Several methods of scaling have been presented in the literature [15-21]. Here we discuss a few of them.

Upper-bound scaling. Examine Figure 14-16. To limit the signal at any point V_i^q to 1, we may select an input scaling factor λ_i such that

$$V_i^q(z) = \lambda_i X(z) F_i(z) \quad (14-222)$$

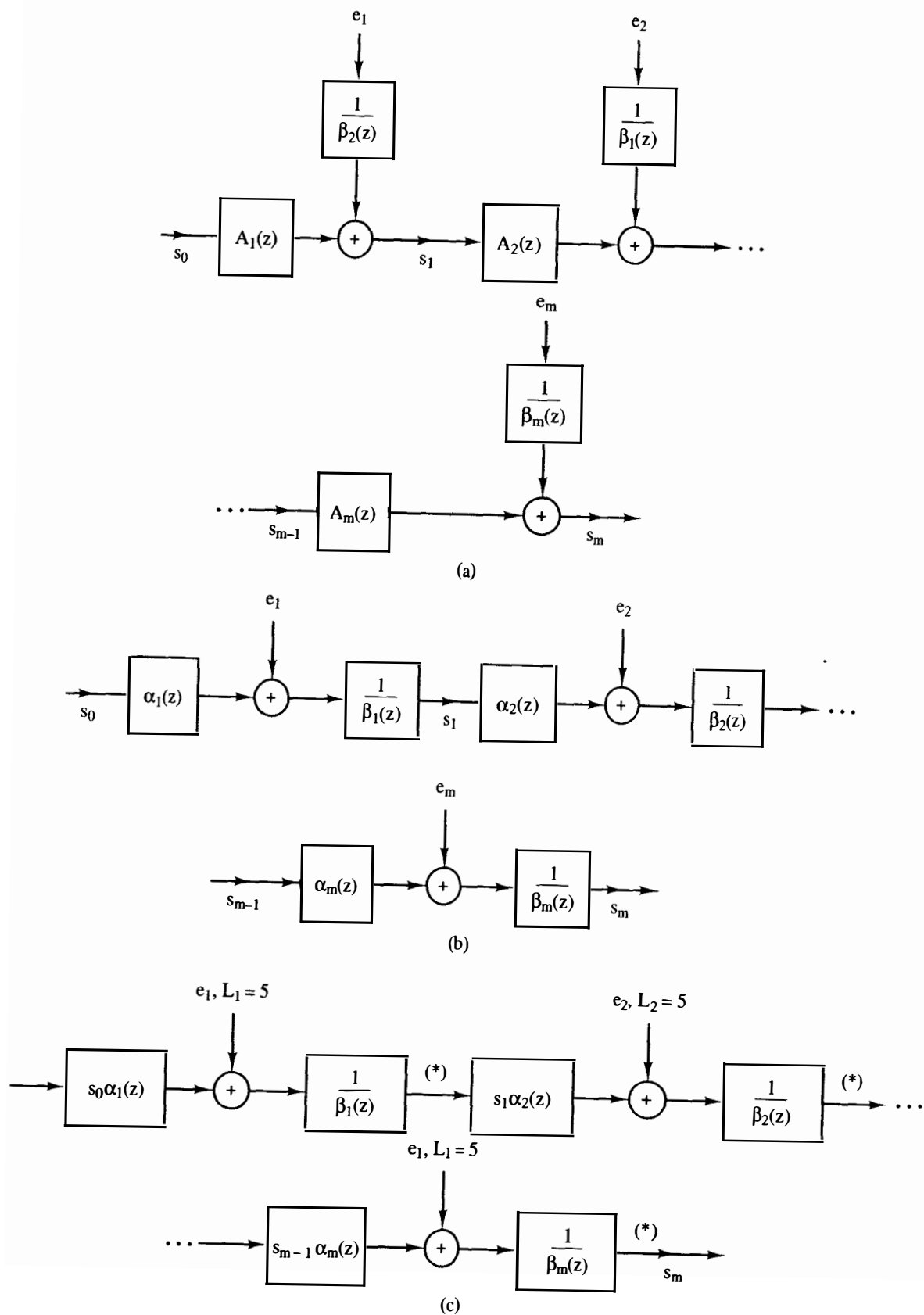


Figure 14-28 2D or 3D cascaded filter: (a) scaling between modules; (b) rearranging terms; (c) numerator scaling.

Hence

$$v_i^q(n) = \sum_{k=0}^{\infty} \lambda_i f_i(k) x(n-k) \quad (14-223)$$

But

$$|v_i^q(n)| \leq \sum_{k=0}^{\infty} |\lambda_i f_i(k)| |x(n-k)| \quad (14-224)$$

and since $|x(n-k)| \leq 1$ by definition,

$$|v_i^q(n)| < \sum_{k=0}^{\infty} |\lambda_i f_i(k)| \quad (14-225)$$

The object of scaling is to ensure that

$$|v_i^q(n)| \leq 1 \quad (14-226)$$

Consequently, we may ensure (14-226) by forcing

$$|v_i^q(n)| \leq \sum_{k=0}^{\infty} |\lambda_i f_i(k)| = 1 \quad (14-227)$$

or

$$\lambda_i = \frac{1}{\sum_{k=0}^{\infty} |f_i(k)|} \quad (14-228)$$

This scaling procedure produces a set of $\lambda_i, i = 1, Q$, where Q is the number of internal quantizers in the structure. If one chooses the scaling constant s :

$$s = \min(\lambda_i), \quad i = 1, Q \quad (14-229)$$

overflow can *never* occur in the filter structure. Equation (14-228) is an absolute upper bound case, and is too conservative in most applications. The signal levels are quite restricted, leaving much of the filter dynamic range unused.

L_p -norm scaling. The inverse z -transform of (14-222) gives the expression

$$v_i^q(n) = \frac{1}{2\pi j} \oint \lambda_i X(z) F_i(z) z^{n-1} dz \quad (14-230)$$

If the contour of integration is the unit circle, $z = e^{j\omega T}$, and

$$v_i^q(n) = \frac{1}{\omega_s} \int_0^{\omega_s} \lambda_i X(e^{j\omega T}) F_i(e^{j\omega T}) e^{jn\omega T} d\omega \quad (14-231)$$

Using the Schwarz inequality [15]

$$\begin{aligned} |v_i^q(n)| &\leq \left[\frac{1}{\omega_s} \int_0^{\omega_s} |\lambda_i F_i(e^{j\omega T})|^2 d\omega \right]^{1/2} \\ &\quad \times \left[\frac{1}{\omega_s} \int_0^{\omega_s} |X(e^{j\omega T})|^2 d\omega \right]^{1/2} \end{aligned} \quad (14-232)$$

Another way to write (14-230) is

$$|v_i^q(n)| \leq \frac{1}{\omega_s} \int_0^{\omega_s} |X(e^{j\omega T})| |\lambda_i F_i(e^{j\omega T})| d\omega$$

or

$$|v_i^q(n)| \leq \left[\max_{\omega \in [0, \omega_s]} |X(e^{j\omega T})| \right] \frac{1}{\omega_s} \int_0^{\omega_s} |\lambda_i F_i(e^{j\omega T})| d\omega \quad (14-233)$$

Similarly,

$$|v_i^q(n)| \leq \left[\max_{\omega \in [0, \omega_s]} |\lambda_i F_i(e^{j\omega T})| \right] \frac{1}{\omega_s} \int_0^{\omega_s} |X(e^{j\omega T})| d\omega \quad (14-234)$$

Equations (14-231)–(14-234) may be conveniently expressed in terms of L_p norms. The L_p norm of a periodic function $G(\omega)$ with period ω_s is

$$\|G\|_p = \left[\frac{1}{\omega_s} \int_0^{\omega_s} |G(\omega)|^p d\omega \right]^{1/p}, \quad p \geq 1 \quad (14-235)$$

If $G(\omega)$ is continuous, then

$$\begin{aligned} \lim_{p \rightarrow \infty} \|G\|_p &= \|G\|_\infty = \max_{\omega \in [0, \omega_s]} |G(\omega)| \\ \|G\|_\infty &\geq \|G\|_p \end{aligned} \quad (14-236)$$

Substituting (14-235) and (14-236) into (14-231)–(14-234) produces

$$|v_i^q(n)| \leq \|\lambda_i F_i\|_2 \|X\|_2 \quad (14-237)$$

$$|v_i^q(n)| \leq \|X\|_\infty \|\lambda_i F_i\|_1 \quad (14-238)$$

$$|v_i^q(n)| \leq \|\lambda_i F_i\|_\infty \|X\|_1 \quad (14-239)$$

In general [21],

$$|v_i^q(n)| \leq \|X\|_p \|\lambda_i F_i\|_q \quad (14-240)$$

where $1/p + 1/q = 1$.

These relations may be used in scaling since (14-240) holds for all X and F_i ; let $\lambda_i F_i = 1$. Then

$$|v_i^q(n)| = |x(n)| \leq \|X\|_p \|1\|_q$$

or

$$|x(n)| = \|X\|_p \quad (14-241)$$

We may add the absolute upper bound

$$|x(n)| \leq \|X\|_p \leq 1 \quad (14-242)$$

But

$$|v_i^q(n)| \leq \|X\|_p \|\lambda_i F_i\|_q \leq 1 \|\lambda_i F_i\|_q \quad (14-243)$$

If we choose λ_i such that

$$|v_i^q(n)| \leq 1 \quad (14-244)$$

then

$$\|\lambda_i F_i\|_q = 1$$

or

$$\lambda_i = \frac{1}{\|F_i\|_q}, \quad q \geq 1 \quad (14-245)$$

These values of λ_i may be used in (14-229). When $q = \infty$, then

$$\lambda_i = \frac{1}{\max_{\omega \in [0, \omega_s]} F_i(e^{j\omega T})} \quad (14-246)$$

which represents sinusoidal scaling. That is, a unit sine-wave input at a frequency that produces $\max F_i(e^{j\omega T})$ will give a unit sine wave out of the filter.

When $q = 2$, then

$$\lambda_i = \frac{1}{\|F_i\|_2} = \left[\frac{1}{\omega_s} \int_0^{\omega_s} |F_i(e^{j\omega T})|^2 d\omega \right]^{-1/2}$$

and

$$\lambda_i^2 = \left[\frac{1}{\omega_s} \int_0^{\omega_s} F(e^{j\omega T}) F(e^{-j\omega T}) d\omega \right]^{-1}$$

but

$$\sum_{k=0}^n (v_i^q(k))^2 = \left(\sum_{k=0}^n x^2(k) \right) \left(\frac{1}{\omega_s} \int_0^{\omega_s} F(e^{j\omega T}) F(e^{-j\omega T}) d\omega \right)$$

or

$$\lambda_i^2 = \frac{\sum_{k=0}^n x^2(k)}{\sum_{k=0}^n (v_i^q(k))^2} \quad (14-247)$$

and the scaling constant relates the mean-squared values of the input and internal variable (energy scaling).

Unit step scaling. If the input $x(n)$ to the filter is a unit step (step of maximum amplitude), then (14-223) becomes

$$v_i^q(n) = \sum_{k=0}^n \lambda_i f_i(k) \quad (14-248)$$

and

$$|v_i^q(n)| \leq \max_{n=[0,\infty]} \left| \sum_{k=0}^n \lambda_i f_i(k) \right| \leq 1 \quad (14-249)$$

Hence we may choose

$$\lambda_i = \left[\max_{n=[0,\infty]} \left| \sum_{k=0}^n f_i(k) \right| \right]^{-1} \quad (14-250)$$

Averaging method [22]. This scaling method attempts to avoid overflow and maximize the signal-to-noise ratio. Let

$$F_{k\max} = \max_{\omega=[0,\omega_s]} |F_{ki}(e^{j\omega T})| \quad (14-251)$$

for the i th signal variable (constraint point) of the k th module in a cascade. Also, let

$$F_{k\min} = \min_{\omega=[0,\omega_s]} |F_{ki}(e^{j\omega T})| \quad (14-252)$$

To avoid overflow one would like to have

$$s_0 F_{1\max} = s_1 F_{2\max} = s_2 F_{3\max} = \cdots = s_{m-1} F_{m\max} \quad (14-253)$$

To maximize the signal-to-noise ratio, however,

$$s_0 F_{1\min} = s_1 F_{2\min} = \cdots = s_{m-1} F_{m\min} \quad (14-254)$$

would be more appropriate. A compromise solution is to average the two:

$$a = \frac{F_{k\max} + F_{k\min}}{2} s_{k-1} \quad (14-255)$$

where a is average for all k . Consequently,

$$\prod_{k=1}^m \left(\frac{F_{k\max} + F_{k\min}}{2} \right) s_{k-1} = a^m \quad (14-256)$$

But by (14-216)

$$\prod_{k=0}^m s_k = a_0$$

and

$$\prod_{k=1}^m s_{k-1} = \frac{a_0}{s_m} \quad (14-257)$$

Therefore, scale factor s_{i-1} may be calculated by

$$a = \frac{F_{i\max} + F_{i\min}}{2} s_{i-1}$$

$$\left(\frac{F_{i\max} + F_{i\min}}{2} s_{i-1} \right)^m = \prod_{k=1}^m \left(\frac{F_{k\max} + F_{k\min}}{2} \right) s_{k-1}$$

$$= \frac{\prod_{k=1}^m s_{k-1} \prod_{k=1}^m (F_{k\max} + F_{k\min})}{2^m}$$

or

$$s_{i-1} = \frac{\left[(a_0/s_m) \prod_{k=1}^m (F_{k\max} + F_{k\min}) \right]^{1/m}}{F_{i\max} + F_{i\min}} \quad (14-258)$$

for $i = 1, m$.

Optimization

Equation (14-219) expresses the output noise of a cascade of 1D second-order modules. In order to minimize the output noise, several researchers have used optimization techniques [16-20]. The scaling method is first selected. For example, if L_2 -norm scaling is used in $q = 2$. In Figure 14-27, (14-245) is applicable with $q = 2$. In Figure 14-27, points at which overflow must be constrained are labeled (*). So, for the l th module,

$$s_0 s_1 s_2 \cdots s_{l-1} = \frac{1}{\|A_1 A_2 \cdots A_{l-1} / \beta_l\|_2} \quad (14-259)$$

or

$$(s_0 \cdots s_{l-1}) = \left\| \frac{A_1 \cdots A_{l-1}}{\beta_l} \right\|_2 = 1$$

$$(s_0 \cdots s_{l-1})^2 \left[\frac{1}{\omega_s} \int_0^{\omega_s} \left| \frac{A_1 \cdots A_{l-1}}{\beta_l} \right|^2 d\omega \right] = 1 \quad (14-260)$$

Consequently,

$$\frac{1}{2\pi j} \oint \prod_{i=1}^l s_{i-1}^2 \frac{\alpha_{i-1}(z) \alpha_{i-1}(1/z)}{\beta_i(z) \beta_i(1/z)} \frac{dz}{z} = 1 \quad (14-261)$$

where $\alpha_0(z) = 1$. Finally, we may incorporate (14-261) into (14-219):

$$\sigma_{e_n}^2 = \frac{2^{-2b}}{2} \left[L_m + \sum_{l=1}^m \frac{L_{l-1}}{2\pi j} \oint \prod_{i=1}^l s_{i-1}^2 \frac{\alpha_{i-1}(z) \alpha_{i-1}(1/z)}{\beta_i(z) \beta_i(1/z)} \frac{dz}{z} \right. \\ \left. \times \frac{1}{2\pi j} \oint \prod_{i=l}^m s_i^2 \frac{\alpha_i(z) \alpha_i(1/z)}{\beta_i(z) \beta_i(1/z)} \frac{dz}{z} \right] \quad (14-262)$$

This relation is the one used in optimization.

Most designers of real-time digital filters do not have optimization programs [23] available. Consequently, in what follows we present design guidelines that usually give “good” results, but not optimal. One should always remember that an optimal solution to (14-219) involves selecting a particular scaling method as we did

in (14-262). The selection of an “optimal” scaling method will depend on the application. For example, in closed-loop control systems, signal levels into a digital controller can be very low. In these cases L_∞ or absolute upper bound scaling would not be wise. The L_2 , L_1 , unit step, or averaging methods would be better.

Pole–Zero Pairing

In optimizing (14-219), a search of all possible pairing of numerator and denominator terms requires $(m!)^2$ evaluations of (14-219). Jackson [24] has suggested that good results are obtained if a pairing is selected that minimizes $\|A_i\|_\infty$ for all i . From (14-236),

$$\|A_i\|_\infty = \max_{\omega \in [0, \infty]} |A_i(\omega)| \quad (14-263)$$

We may minimize $\|A_i\|_\infty$ by minimizing the peak frequency response of A_i . The following graphical procedure, though not optimal, does give good results by minimizing the peak frequency response of pole pairs that are near the unit circle:

1. Plot poles and zeros graphically in the z -plane.
2. Pair real poles with each other. Find the real pole nearest the $z = +1$ point. Pair it with the real pole farthest from the $z = +1$ point. Continue until all real poles have been paired.
3. Pair poles and zeros. Begin by finding the pole nearest the unit circle. Match it with the zero nearest its location. If the zero is real, match the other pole of the pole pair in the same manner with the nearest real zero. Repeat step 3 until all poles and zeros have been matched.

Example 14.2

Consider the fourth-order digital filter of Figure 14-29. Here the real poles are first paired. Then the complex pair of poles is matched with the complex zeros. Finally, the real poles are matched with the real zeros.

Example 14.3

Lee [17] has optimized the cascade realization of a cascaded 1D seventh-order filter:

$$\begin{aligned} \alpha_1: & 1 + 1.12368z^{-1} + z^{-2} \\ \beta_1: & 1 + 0.05358156z^{-1} + 0.9403549z^{-2} \\ \alpha_2: & 1 + 0.216853z^{-1} + z^{-2} \\ \beta_2: & 1 + 0.019248z^{-1} + 0.74403z^{-2} \\ \alpha_3: & 1 + 0.42371z^{-1} + z^{-2} \\ \beta_3: & 1 + 0.19405z^{-1} + 0.33908z^{-2} \\ \alpha_4: & 1 + z^{-1} \\ \beta_4: & 1 + 0.16535z^{-1} \end{aligned} \quad (14-264)$$

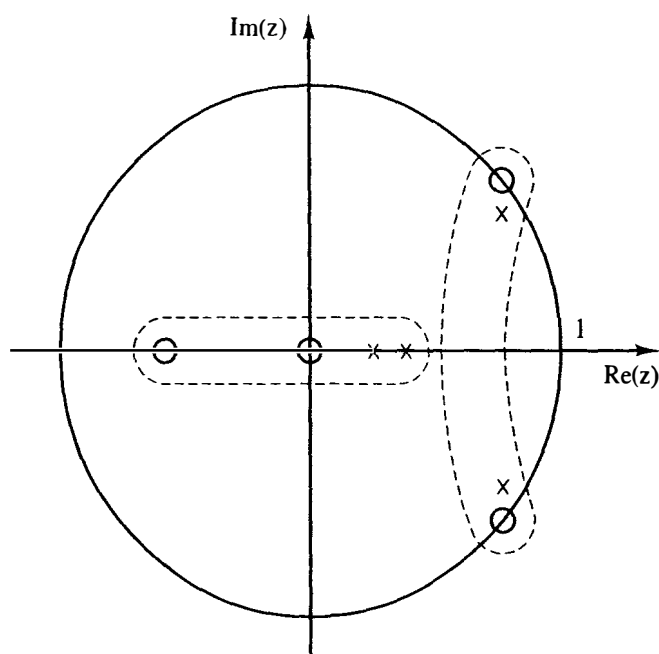


Figure 14-29 Pole-zero pairing.

Consider the pole-zero plot of Figure 14-30 and the following algorithm.

1. Real pole β_4 is matched with α_4 .
2. Complex poles β_1 are matched with complex zeros α_2 .
3. Complex poles β_2 are matched with complex zeros α_3 .
4. Complex poles β_3 are matched with complex zeros α_1 .

This pairing is the same as obtained by optimization in Ref. 17.

Ordering

Once the second-order modules have been paired, they must be ordered to minimize the output noise and limit cycle response. Ordering algorithms have been proposed in the literature [24–27]. Here we will derive a method based on (14-219) for cascaded 1D modules. The technique may be extended to other direct structures.

Since

$$\begin{aligned} \frac{1}{2\pi j} \oint A_i(z) A_i(1/z) dz/z &= \frac{1}{\omega_s} \int_0^{\omega_s} |A_i(e^{j\omega T})|^2 d\omega \\ &= \sum_{m=0}^{\infty} a_i^2(m) \end{aligned} \quad (14-265)$$

we may write (14-219) in the form

$$\sigma_{e_n}^2 = \frac{2^{-2b}}{12} \left[L_m + \sum_{l=1}^m L_{l-1} \frac{1}{\omega_s} \int_0^{\omega_s} \left| \prod_{i=l}^m s_i A_i(e^{j\omega T}) \right|^2 d\omega \right] \quad (14-266)$$

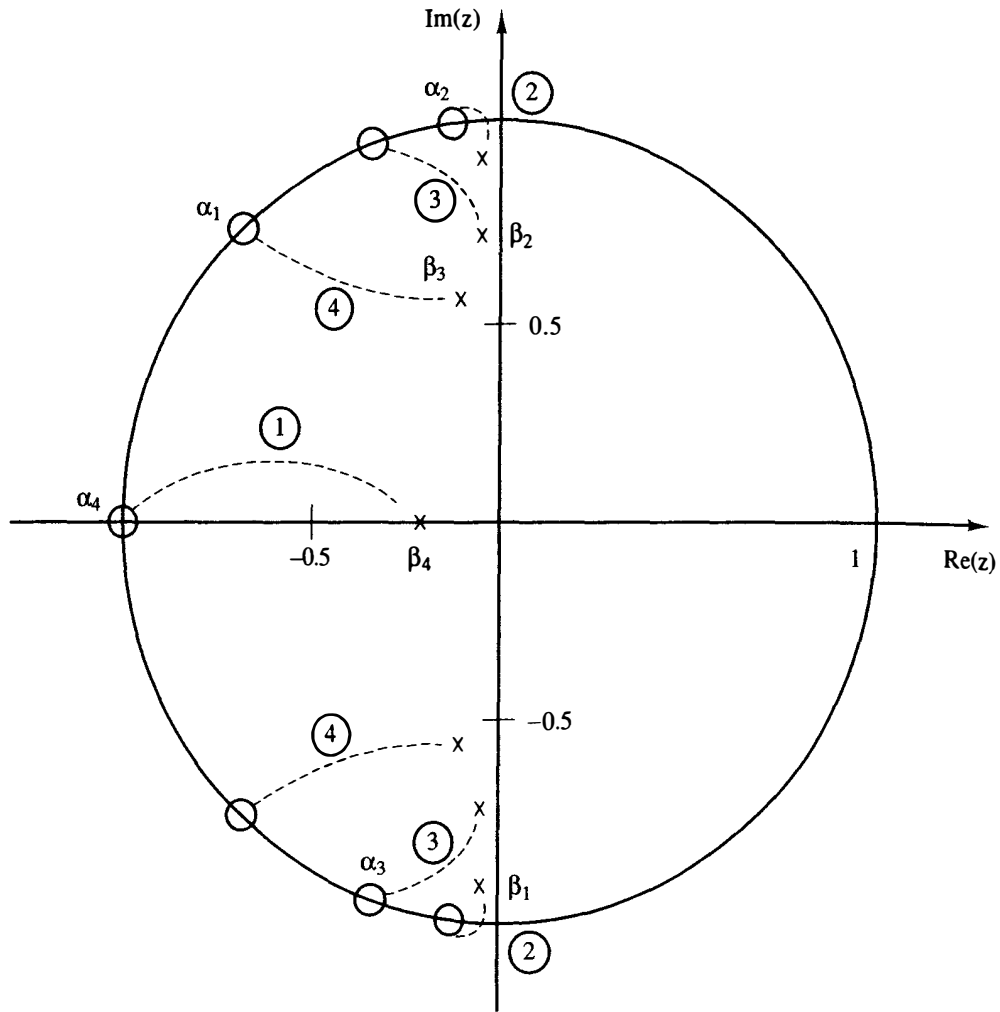


Figure 14-30 Pole-zero pairing example.

Consequently,

$$\sigma_{e_n}^2 = \frac{2^{-2b}}{12} \left[L_m + \sum_{l=1}^m L_{l-1} \frac{1}{\omega_s} \int_0^{\omega_s} \prod_{i=l}^m s_i^2 |A_i(e^{j\omega T})|^2 d\omega \right] \quad (14-267)$$

But we may interchange the integral and product and create an inequality:

$$\begin{aligned} \sigma_{e_n}^2 &\leq \frac{2^{-2b}}{12} \left[L_m + \sum_{l=1}^m L_{l-1} \prod_{i=l}^m \frac{1}{\omega_s} \int_0^{\omega_s} s_i^2 |A_i(e^{j\omega T})|^2 d\omega \right] \\ &\leq \frac{2^{-2b}}{12} \left[L_m + \sum_{l=1}^m L_{l-1} \prod_{i=l}^m C_i \right] \end{aligned} \quad (14-268)$$

where

$$C_i = \frac{1}{\omega_s} \int_0^{\omega_s} s_i^2 |A_i(e^{j\omega T})|^2 d\omega = \sum_{m=0}^{\infty} s_i^2 a_i^2(m)$$

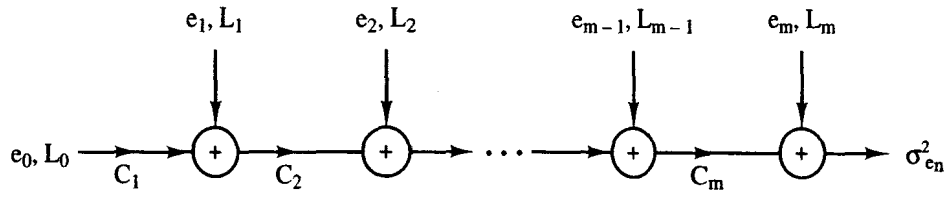


Figure 14-31 Ordering model for 1D modules.

is a constant for each module. Equation (14-268) may be expanded:

$$\sigma_{e_n}^2 \leq \frac{2^{-2b}}{12} [L_m + L_{m-1}C_m + L_{m-2}C_{m-1}C_m + \cdots + L_0C_1C_2\cdots C_m] \quad (14-269)$$

and may also be written as

$$\sigma_{e_n}^2 \leq (C_m \cdots (C_2(C_1L_0 + L_1) + L_2) \cdots + L_m) \frac{2^{-2b}}{12} \quad (14-270)$$

The result in (14-270) may be drawn graphically as shown in Figure 14-31. Since $L_0 = 2$, $L_m = 3$ and $L_i = 5$, $\sigma_{e_n}^2$ may be reduced in value by requiring that

$$C_1 \geq C_2 \geq C_3 \cdots \geq C_m \quad (14-271)$$

These constants may be calculated for each module once the pairing and scaling have been accomplished.

Design Guidelines

When a designer is faced with implementing a cascade of second-order modules, the following guidelines for the cascade procedure may be used to achieve a “good” suboptimal implementation.

1. Factor $D(z)$ of (14-212) to obtain (14-213).
2. Quantize the coefficients for the desired implementation wordlength (e.g., 16 bits for the Intel 8086).
3. Verify that (14-213) with quantized coefficients meets the system specifications.
4. Employ the pole-zero pairing algorithm.
5. Choose a structure for the modules (e.g., 1D).
6. Scaling may now be applied to each module *independently*. We suggest the unit step method (except for integrators). However, any of the other methods may be used.
7. Apply the ordering algorithm. This algorithm uses constants which may be calculated from each independent module. The implementation is now complete.

8. Simulate the implementation in the open-loop case and test its step response to assure that the dynamic range of the internal variables is midrange.
9. Simulate the implementation in the closed-loop case to ensure that system specifications are met.

The key to the foregoing procedure is that calculations are done on independent modules to determine scaling and ordering. This drastically reduces the computations needed to implement a filter, and can be done by hand without computer-aided design programs.

14.8 PARALLEL SECOND-ORDER MODULES

In implementing (14-212) as a parallel connection of second-order modules, $D(z)$ may be expressed, using a partial-fraction expansion, as

$$D(z) = \beta_0 + \sum_{i=1}^m \beta_i(z) \quad (14-272)$$

where

$$B_i(z) = \frac{\beta_{i1} z^{-1} + \beta_{i2} z^{-2}}{1 + \beta_{i3} z^{-1} + \beta_{i4} z^{-2}} = \frac{\alpha_i(z)}{\beta_i(z)}$$

Figure 14-32a depicts the parallel realization. The modules B_i may be implemented using the direct structures of Figure 14-26. The 1D (or 4D) direct structure has been used in the implementation of Figure 14-32b. The scaling constants have been added to avoid overflow in the second-order modules. The inverse of the scaling constant is used to restore the signal level before the output adder is reached, as is required in (14-272). Here the error variance may be expressed, by using (14-217),

$$\sigma_{e_n}^2 = \frac{2^{-2b}}{12} \left[3m + \sum_{i=1}^m \frac{2m}{2\pi j} \oint \frac{B_i(z) B_i(1/z)}{s_i^2} \frac{dz}{z} \right] \quad (14-273)$$

and by (14-265),

$$\sigma_{e_n}^2 = \frac{2^{-2b}}{12} \left[3m + \sum_{i=1}^m \frac{2m}{s_i^2} \sum_{l=0}^{\infty} b_i^2(l) \right] \quad (14-274)$$

Implementing (14-272) in the 2D or 3D structures produces the error configuration shown in Figure 14-32c. The output noise variance in this case is computed by

$$\sigma_{e_n}^2 = \frac{2^{-2b}}{12} \left[\sum_{i=1}^m \frac{L_m}{2\pi j} \oint \frac{1}{s_i^2 \beta_i(z) \beta_i(1/z)} \frac{dz}{z} \right] \quad (14-275)$$

where $L_m = 5$.

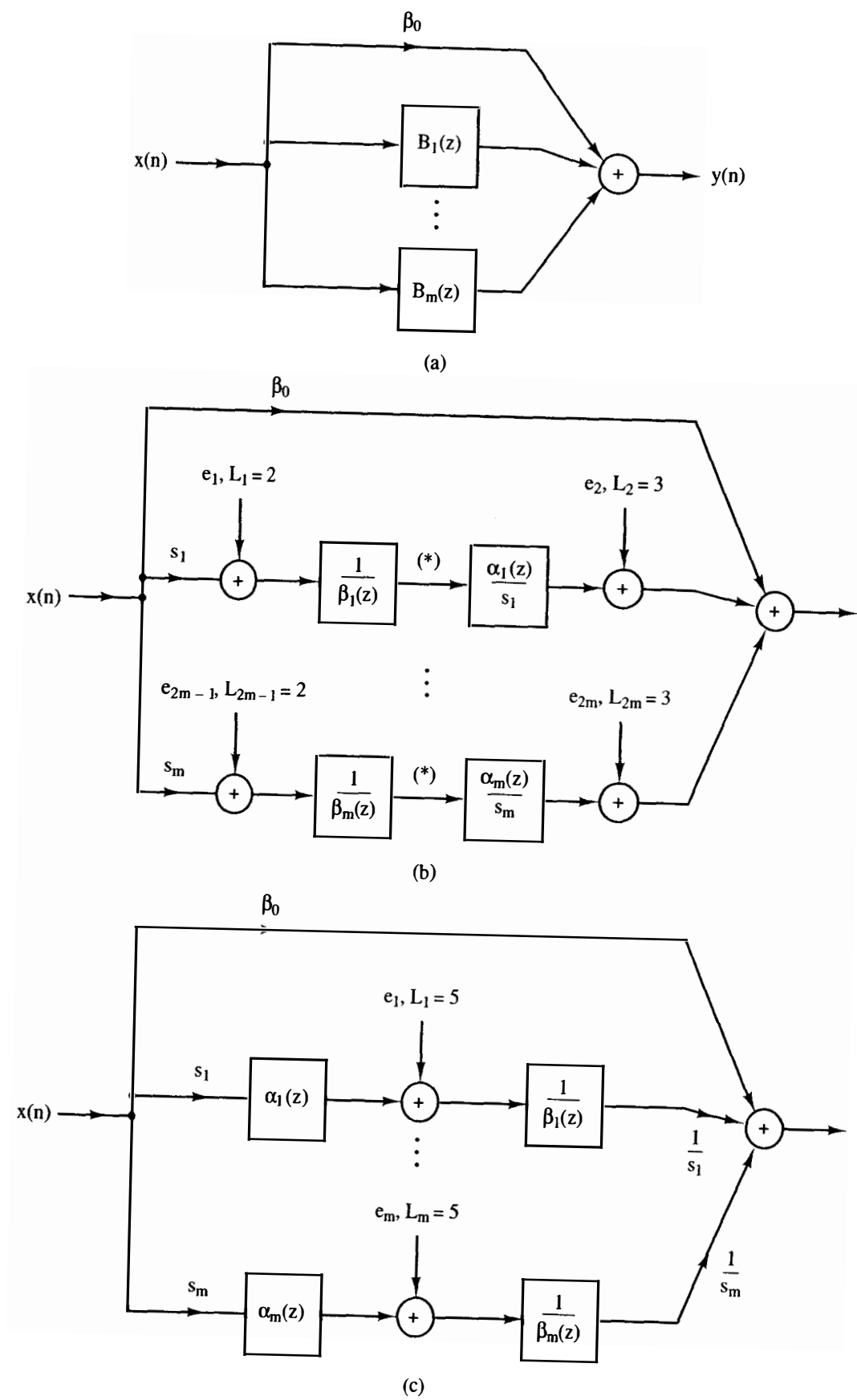


Figure 14-32 Parallel implementation models: (a) general model; (b) 1D, 4D implementation; (c) 2D, 3D implementations.

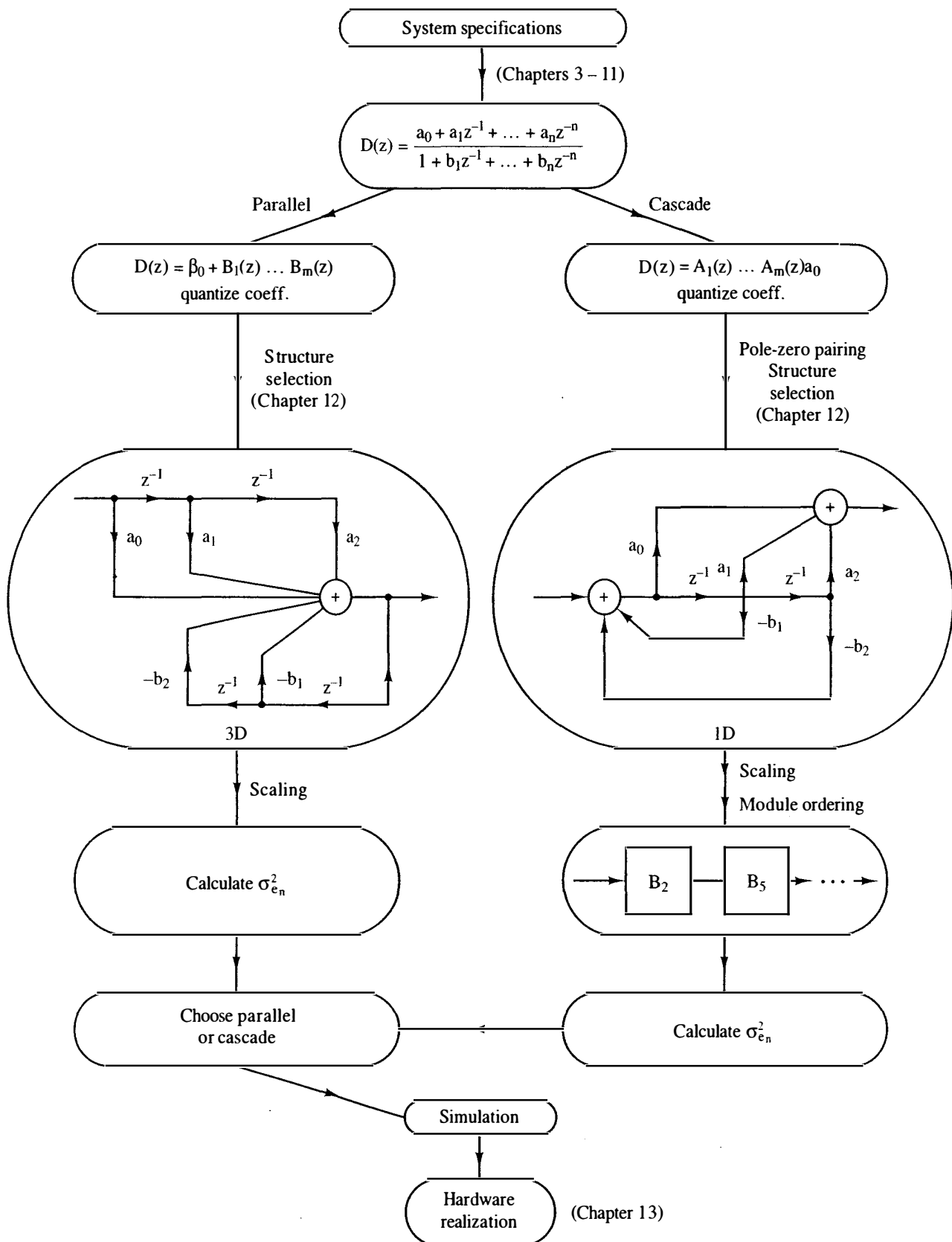


Figure 14-33 General design procedure.

The parallel implementation of a digital filter is a rather straightforward process.

1. Perform a partial-fraction expansion on (14-212) to obtain (14-272).
2. Quantize the coefficients to the realizable wordlength.
3. Verify that (14-272) with quantized coefficients meets the system specifications.
4. Choose a structure for the second-order modules.
5. Scaling may be applied to each module (e.g., the unit step method). The implementation is now complete.
6. Simulate the open-loop digital filter and test its step, impulse, and sinusoidal responses to assure that the dynamic range of all variables is appropriate.
7. Insert the digital filter simulation into the total system simulation to ensure that system specifications have been satisfied.

Note that the design procedure for the parallel case is similar to the cascade case, with the exception that pairing and ordering are not necessary.

14.9 SUMMARY

In this chapter we have discussed the finite-wordlength complications of implementing digital filters. Signal amplitude quantization was shown to add low-level noise to signal variables. The resulting output effect is described by random noise or limit cycles. Coefficient quantization is shown to change the transfer characteristics of the digital filter. Overflow has been shown to impose disastrous consequences on the digital filter; hence overflow oscillations must be avoided.

We have also discussed the cascade and parallel implementations of digital filters. In the cascade case we factored $D(z)$ and used coefficient quantization, pole-zero pairing, direct structures for second-order modules, module scaling, module ordering, and simulation to achieve and verify our implementation. In the parallel case we used a partial-fraction expansion of $D(z)$ and omitted the pole-zero pairing and ordering steps. The overall design process is summarized in Figure 14-33.

REFERENCES

1. A. B. Sripad and D. L. Snyder, "Quantization Errors in Floating-Point Arithmetic," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-26, pp. 456–463, Oct. 1978.
2. R. E. Crochiere and A. V. Oppenheim, "Analysis of Linear Digital Networks," *Proc. IEEE*, Vol. 63, pp. 581–595, Apr. 1975.
3. B. Gold and C. M. Radar, *Digital Processing of Signals*. New York: McGraw-Hill Book Company, 1969.

4. S. K. Mitra, K. Hirano, and H. Sakaguchi, "A Simple Method of Computing the Input Quantization and Multiplication Roundoff Errors in a Digital Filter," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-22, pp. 326–329, Oct. 1974.
5. L. B. Jackson, "An Analysis of Limit Cycles Due to Multiplication Rounding in Recursive (Sub) Filters," *Proc. 7th Annu. Allerton Conf. Circuit Syst. Theory*, 1969, pp. 69–78.
6. S. R. Parker and S. F. Hess, "Limit-Cycle Oscillations in Digital Filters," *IEEE Trans. Circuit Theory*, Vol. CT-18, pp. 687–697, Nov. 1971.
7. I. W. Sandberg and J. F. Kaiser, "A Bound on Limit Cycles in Fixed-Point Implementations of Digital Filters," *IEEE Trans. Audio Electroacoust.*, Vol. AU-20, pp. 110–112, June 1972.
8. T. A. C. M. Claasen, W. F. G. Mecklenbraeuer, and J. B. H. Peek, "Some Remarks on the Classification of Limit Cycles in Digital Filters," *Phillips Res. Rep.*, Vol. 28, pp. 297–305, Aug. 1973.
9. J. L. Long and T. N. Trick, "An Absolute Bound on Limit Cycles Due to Roundoff Errors in Digital Filters," *IEEE Trans. Audio Electroacoust.*, Vol. AU-21, pp. 27–30, Feb. 1973.
10. T. Claasen, W. F. G. Mecklenbraeuer, and J. B. H. Peek, "Frequency Domain Criteria for the Absence of Zero-Input Limit Cycles in Nonlinear Discrete-Time Systems, with Applications to Digital Filters," *IEEE Trans. Circuits Syst.*, Vol. CAS-22, pp. 232–239, Mar. 1975.
11. V. B. Lawrence and K. V. Mina, "Control of Limit Cycle Oscillations in Second-Order Recursive Digital Filters Using Constrained Random Quantization," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-26, pp. 127–134, Apr. 1978.
12. L. B. Jackson, "Limit Cycles in State-Space Structures for Digital Filters," *IEEE Trans. Circuits Syst.*, Vol. CAS-26, pp. 67–68, Jan. 1979.
13. P. M. Ebert, J. E. Mazo, and M. G. Taylor, "Overflow Oscillations in Digital Filters," *Bell Syst. Tech. J.*, Vol. 48, pp. 2999–3020, Nov. 1969.
14. W. L. Mills, C. T. Mullis, and R. A. Roberts, "Digital Filter Realizations without Overflow Oscillations," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-24, pp. 334–338, Aug. 1978.
15. L. B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," *Bell Syst. Tech. J.*, Vol. 49, pp. 159–184, Feb. 1970.
16. S. Y. Hwang, "On Optimization of Cascade Fixed-Point Digital Filters," *IEEE Trans. Circuits Syst.*, Vol. CAS-21, pp. 163–166, Jan. 1974.
17. W. S. Lee, "Optimization of Digital Filters for Low Roundoff Noise," *IEEE Trans. Circuits Syst.*, Vol. CAS-22, pp. 424–431, May 1974.
18. B. Liu and A. Peled, "Heuristic Optimization of the Cascade Realization of Fixed-Point Digital Filters," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-23, pp. 464–473, Oct. 1975.
19. E. Lueder, H. Hug, and W. Wolf, "Minimizing the Roundoff Noise in Digital Filters by Dynamic Programming," *Frequenz*, Vol. 29, pp. 211–214, 1975.
20. C. T. Mullis and R. A. Roberts, "Synthesis of Minimum Roundoff Noise Fixed Point Digital Filters," *IEEE Trans. Circuits Syst.*, Vol. CAS-23, pp. 551–562, Sept. 1976.
21. B. P. Gaffney and J. N. Gowdy, "A Symmetry Relationship for "Between" Scaling in Cascade Digital Filters," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-25, pp. 350–351, Aug. 1977.

22. G. Moschytz, *Linear Integrated Networks Fundamentals*. New York: Van Nostrand Reinhold Company, 1974.
23. A. Peled and B. Liu, *Digital Signal Processing: Theory, Design, and Implementation*. New York: John Wiley & Sons, Inc., 1976.
24. L. B. Jackson, "Roundoff Noise Analysis for Fixed Point Digital Filters Realized in Cascade or Parallel Form," *IEEE Trans. Audio Electroacoust.*, Vol. AU-18, pp. 107–122, June 1970.
25. E. P. F. Kan and J. K. Aggarwal, "Minimum-Deadband Design of Digital Filters," *IEEE Trans. Audio Electroacoust.*, Vol. AU-19, pp. 292–296, Dec. 1971.
26. G. A. Maria and M. F. Fahmy, "Limit Cycle Oscillations in a Cascade of First- and Second-Order Digital Sections," *IEEE Trans. Circuits Syst.*, Vol. CAS-22, pp. 131–134, Feb. 1975.
27. K. Steiglitz and B. Liu, "An Improved Algorithm for Ordering Poles and Zeros of Fixed-Point Recursive Digital Filters," *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-24, pp. 341–343, Aug. 1976.

PROBLEMS

14-1. For the 1D structure for second-order modules:

- (a) Find $\partial D(z)/\partial a_i$, $\partial D(z)/\partial b_i$.
- (b) Check the results in part (a) using (14-80).
- (c) Find $\Delta D(z)$

14-2. Assume that

$$D(z) = \frac{1 + 1.858741z^{-1} + z^{-2}}{1 - 0.748386z^{-1} + 0.213374z^{-2}}$$

is to be implemented using the 1D structure and an 8-bit two's-complement number system:

coefficients: (s m.m m m m m m)_{2cns}

- (a) Find Δa_i , Δb_i .
- (b) Find $\Delta D(z)$. *Hint:* Use the results of Problem 14-1.
- (c) Plot $\Delta D(e^{j\omega T})$.

14-3. Repeat Problem 14-2 for a 16-bit two's-complement number system.

14-4. For the direct structures for second-order modules, assume an 8-bit two's-complement number system

coefficient: (s m.m m m m m m)_{2cns}

variables: (s.m m m m m m m)_{2cns}

Calculate the output steady-state error bound if each summing node is implemented using double-precision accumulation before round-off (Q_2 quantizers in Figures 14-14b and 14-21e). Use the $D(z)$ of Problem 14-2. Which structure exhibits the best steady-state error performance?

14-5. For the digital filters below, assume that two's-complement round-off is used to

implement each filter in a direct structure. Are overflow oscillations possible? Are limit cycles possible? If limit cycles are possible, find their bounds.

(a) $D_1(z) = 1/(1 + 0.98381z^{-1} + 0.09443z^{-2})$

(b) $D_2(z) = 1/(1 + 0.17819z^{-1} - 0.17187z^{-2})$

(c) $D_3(z) = 1/(1 - 1.49513z^{-1} + 0.56292z^{-2})$

14-6. Consider the sixth-order filter [24] in cascaded 2D form:

$$\alpha_1: 1 - 1.8118373z^{-1} + z^{-2}$$

$$\beta_1: 1 - 1.7636952z^{-1} + 0.90352914z^{-2}$$

$$\alpha_2: 1 - 1.6545862z^{-1} + z^{-2}$$

$$\beta_2: 1 - 1.4427789z^{-1} + 0.84506679z^{-2}$$

$$\alpha_3: 1 - 1.7442502z^{-1} + z^{-2}$$

$$\beta_3: 1 - 1.5334490z^{-1} + 0.75829007z^{-2}$$

Use the graphical procedure of Example 14.3 to find a good pole-zero pairing.

14-7. Repeat Problem 14-6 for the following ninth-order filter [17]:

$$\alpha_1 = 1 + 1.12z^{-1} + z^{-2}$$

$$\beta_1 = 1 + 0.01781z^{-1} + 0.97976z^{-2}$$

$$\alpha_2 = 1 + 0.14439z^{-1} + z^{-2}$$

$$\beta_2 = 1 - 0.0093915z^{-1} + 0.90944z^{-2}$$

$$\alpha_3 = 1 + 0.07648z^{-1} + z^{-2}$$

$$\beta_3 = 1 - 0.09486z^{-1} + 0.71287z^{-2}$$

$$\alpha_4 = 1 + 0.3878z^{-1} + z^{-2}$$

$$\beta_4 = 1 - 0.2755z^{-1} + 0.33098z^{-2}$$

$$\alpha_5 = 1 + z^{-1}$$

$$\beta_5 = 1 - 0.20466z^{-1}$$

Case Studies

15.1 INTRODUCTION

This chapter presents three case studies of digital control systems that have been implemented. Two of the control systems were designed using the frequency-response techniques of Chapter 8, and the design of the third, even though empirical, was based on the techniques of Chapter 8.

The first case study is a second-order position control system (servomotor) for which both a phase-lead and a phase-lag controller were designed. The responses are compared. Sampling rate selection is discussed, and certain significant effects from plant nonlinearities are noted. The digital controller is implemented using a Texas Instruments TI9900 microprocessor.

The second system studied is an environmental chamber control system, which is composed of a temperature control system, a carbon dioxide control system, a chamber water-loss monitor, an outside rainfall monitor, and a data acquisition system. The temperature system controller is a PID compensator; the carbon dioxide system controller is a quasi-proportional compensator. Both control systems were designed empirically, and all operations are implemented via a time-shared TI9900 microprocessor. The microcomputer software is discussed.

The third case study is the lateral control system of an automatic aircraft landing system for U.S. Marine fighter aircraft. The plant for this control system is the aircraft lateral dynamics, including the bank autopilot, and is ninth order. The digital controller generates bank commands into the autopilot, and the aircraft position is determined by a phased array radar. The system contains significant noise and disturbance inputs, which must be considered in the controller design. The digital controller is a PID compensator plus added filtering to reduce noise effects.

15.2 SERVOMOTOR SYSTEM

The design of a digitally controlled servomotor system [1] is presented in this section. This system is low order and presents no particular design nor implementation difficulties. However, the system does contain nonlinearities which have an observable influence on system response. These nonlinear effects will be discussed as they are encountered.

The control-system block diagram is shown in Figure 15-1 and the system hardware configuration is given in Figure 15-2. The Texas Instruments TI9900 microprocessor system [2] was chosen for the implementation of the digital controller. At the time of the system design, this microprocessor system was one of the few 16-bit processors available. In addition, the processor has hardware multiplication, and software support is available. The terminal indicated in Figure 15-1 was chosen to be a Texas Instruments Microterminal [2] and is used to initialize the system, change filter parameters if desired, test system operation, and so on.

The data (number) format for data internal to the computer is shown in Figure 15-3 and is fixed-point format. The magnitudes of both the fractional part and the integer part of numbers are presented by 16 bits. Thus the dynamic range is from a (nonzero) minimum of [3]

$$2^{-16} = 0.000015$$

to a maximum of

$$(2^{16} - 1)_{\text{integer}} + (1 - 2^{-16})_{\text{fraction}} = 65,535.999985$$

The 32-bit precision is not needed for this application. It was chosen to facilitate data manipulations, since processing time is abundant. In addition, the sign flag was

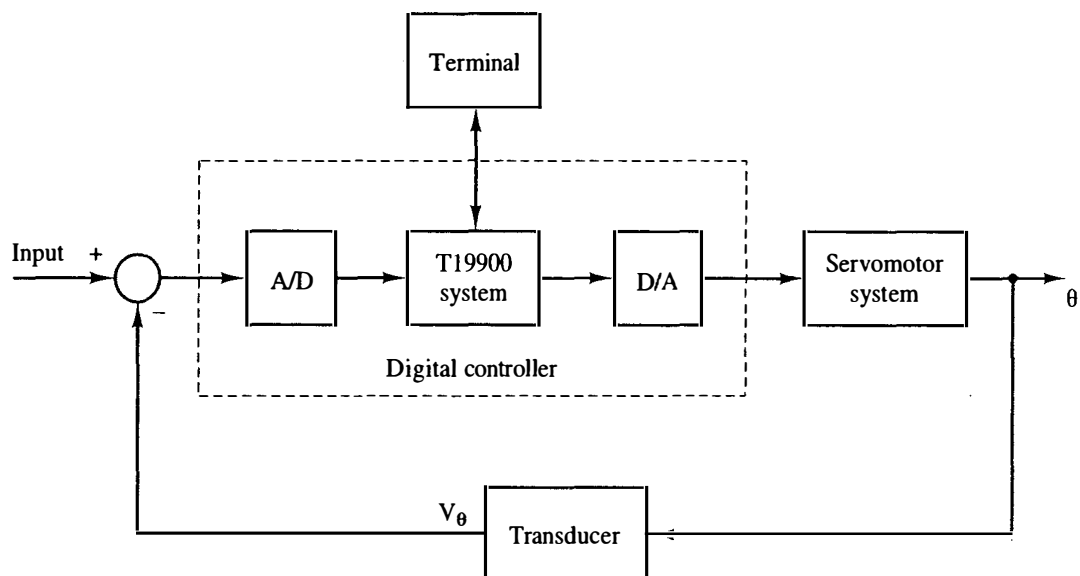


Figure 15-1 System block diagram.

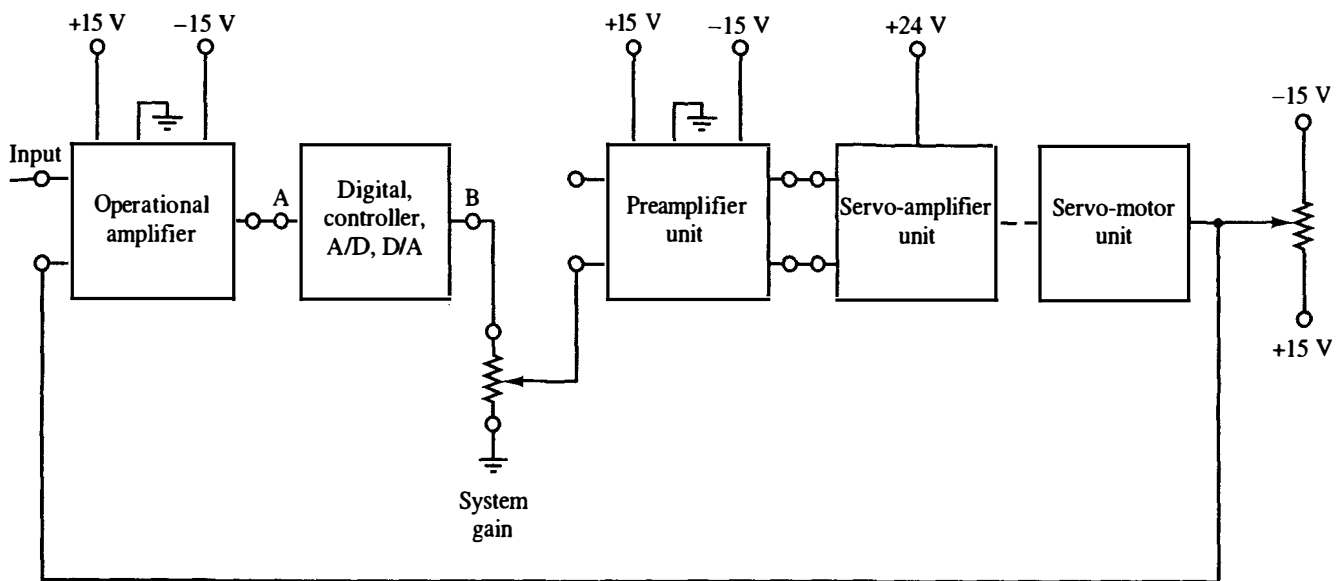


Figure 15-2 System hardware.

placed external to the integer part of the number to facilitate multiplication. The filter coefficients are stored in sign-magnitude format, which is the form required for the multiplication hardware.

System Model

The control system can be mathematically modeled as shown in Figure 15-4. A first step in the design process was to determine the plant transfer function $G_P(s)$. This transfer function was obtained experimentally by removing the digital controller, the A/D, and the D/A shown in Figure 15-2. Thus, in this figure, points A and B were connected. Since the servomotor is dc and armature-controlled, the plant transfer function was assumed to be [4]

$$G_P(s) = \frac{\omega_n^2}{s(s + 2\delta\omega_n)} \quad (15-1)$$

giving a closed-loop transfer function of

$$T(s) = \frac{G_P(s)}{1 + G_P(s)} = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \quad (15-2)$$

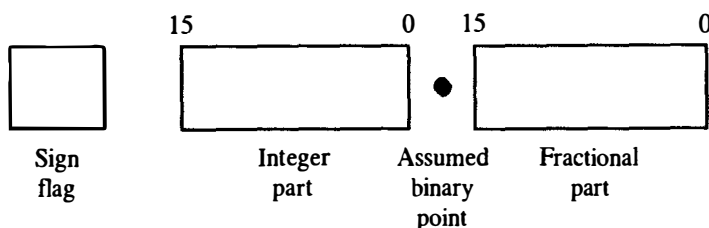


Figure 15-3 Computer data format.

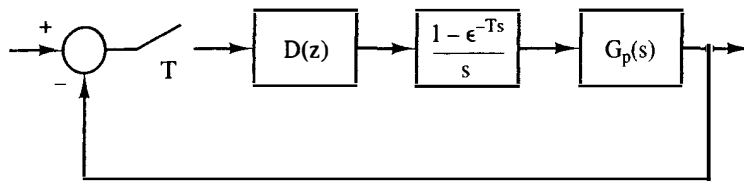


Figure 15-4 Control system model.

Thus the system frequency response is given by

$$T(j\omega) = \frac{\omega_n^2}{(\omega_n^2 - \omega^2) + j2\delta\omega_n\omega} \quad (15-3)$$

To obtain the system model experimentally, a sinusoid was applied to the system and the frequency varied until the system output lagged the input by 90° in phase. As seen from (15-3), this frequency is ω_n . The amplitude of the response at this frequency was then used to calculate δ . The resulting transfer function was found to be

$$T(s) = \frac{36}{s^2 + 3.6s + 36} \quad (15-4)$$

Next a value for the sample period, T , was determined experimentally. One criterion that has been used successfully is to choose T as approximately one-tenth of the system rise time [5,6]. The system step response is recorded in Figure 15-5, indicating a rise time of approximately 0.3 s. Thus a sample period of 30 ms is a value for consideration.

To test this sample period, the system was connected with the microcomputer in the loop, as shown in Figure 15-1. The computer was programmed to be a simple gain of unity (i.e., the A/D, computer, and D/A performed the function of a sampler and zero-order hold). The step response was then run for several values of T in the vicinity of T equal to 30 ms. The results are given in Figure 15-6.

First, note that the response for a sample period of 5 ms. is quite close to that of the analog system (see Figure 15-5). Next note that the system is approaching instability for $T = 40$ ms. However, the small amplitude of the oscillation indicates a nonlinear effect, since generally a linear system oscillation triggered by a step input will have approximately the amplitude of the step. This nonlinear oscillation is the limit cycle as discussed in Chapter 14. An investigation of this system determined

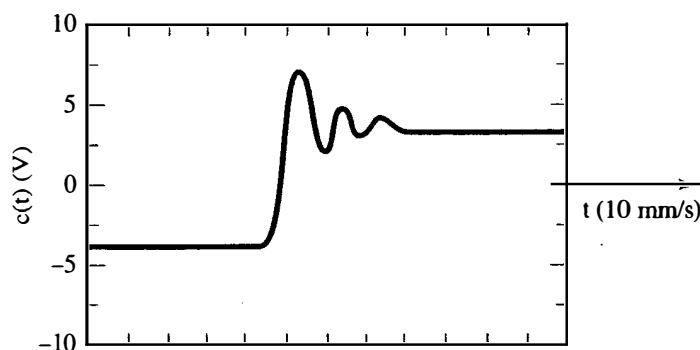


Figure 15-5 Step response of the servomotor system.

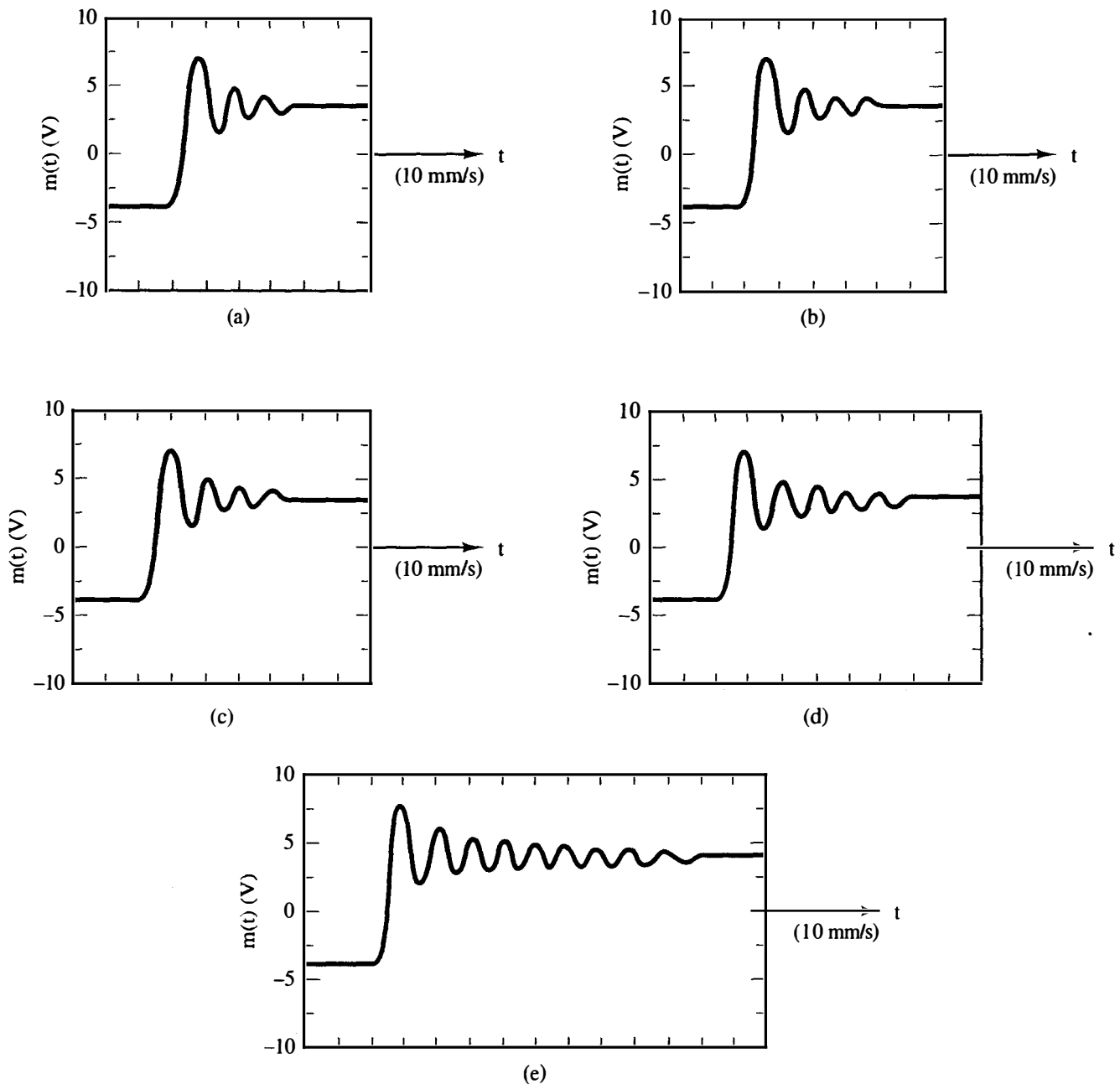


Figure 15-6 Step response of the servomotor for different sampling periods: (a) $T = 5$ ms; (b) $T = 10$ ms; (c) $T = 20$ ms; (d) $T = 30$ ms; (e) $T = 40$ ms.

that a jump resonance nonlinearity [7], usually caused by saturation, is present in the system. Jump resonance has been shown to produce limit cycles in sampled-data systems of this type [8]. Thus, for this system, the sampling rate is determined not by linear sampling theory, but by a system nonlinearity. It should be pointed out that a servo system is often designed such that the servoamplifier is saturated for a large percentage of the time. Thus the motor input voltage is maximum during this time, ensuring maximum speed of response.

Consideration of the step responses in Figure 15-6 resulted in a choice for T of 5 ms. With this value of T , we can calculate $G(z)$. From (15-2) and (15-4),

$$G_P(s) = \frac{36}{s(s + 3.6)} \quad (15-5)$$

Then

$$G(z) = \mathcal{Z} \left[\frac{36(1 - e^{-Ts})}{s^2(s + 3.6)} \right] = \frac{0.00044731z + 0.00044739}{z^2 - 1.982161z + 0.982161} \quad (15-6)$$

The closed-loop transfer function is then

$$\frac{G(z)}{1 + G(z)} = \frac{0.00044731z + 0.00044739}{z^2 - 1.981714z + 0.982608} \quad (15-7)$$

Note the numerical problems. At low frequencies (in the vicinity of $z = 1$), the frequency response is determined principally by the last digit of the numerator coefficients, and thus numerical inaccuracies will be present. These numerical problems are caused by using a sampling frequency that is very large compared to the system natural frequencies. To calculate the frequency response from (15-6) and (15-7), numerical accuracy greater than that shown must be employed. However, use of (7-23) to calculate the frequency response circumvents these problems.

Shown in Figure 15-7 is the measured system frequency response and also the frequency response of an analog (linear) computer simulation of the system. The system frequency response in the vicinity of the resonance peak is actually double valued, because of the jump resonance effects [7], but is shown as being single

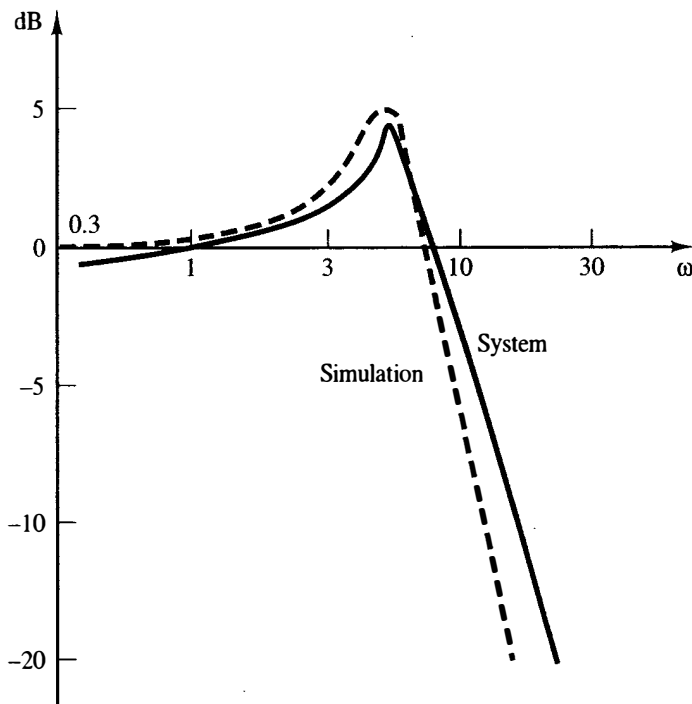


Figure 15-7 Closed-loop frequency response. $D(z) = 1$, $T = 0.005$ s.

valued. Since both of these curves were obtained experimentally, they are plotted versus real, or s -plane, frequency. The w -plane frequency is obtained from (7-10):

$$\omega_w = \frac{2}{T} \tan \frac{\omega T}{2} \quad (15-8)$$

Design

The frequency response for $G(z)$, obtained using (7-23), is plotted in Figure 15-8.

As described in Chapter 8, phase lead design is principally trial and error. Use of the procedure of Chapter 8 resulted in a phase-lead filter transfer function of

$$D(w) = \frac{0.693(1 + w/2.21)}{1 + w/44.8} \quad (15-9)$$

and thus, from (8-14),

$$D(z) = \frac{12.74z - 12.6}{z - 0.798} \quad (15-10)$$

The computer calculates

$$y(nT) = 12.74x(nT) - 12.6x(nT - T) + 0.798y(nT - T)$$

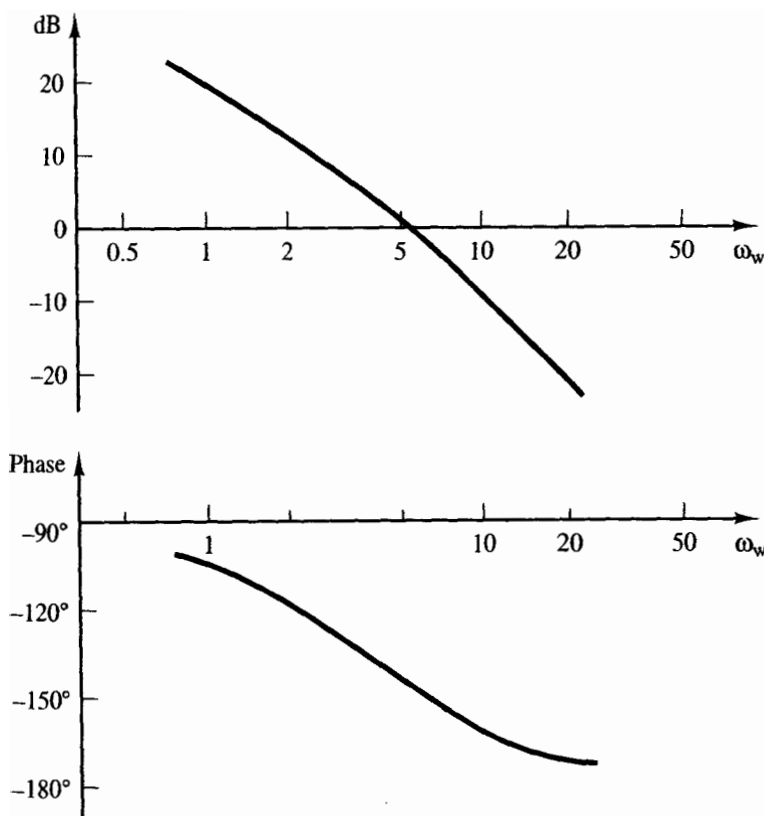


Figure 15-8 Open-loop frequency response. $T = 0.005$ s.

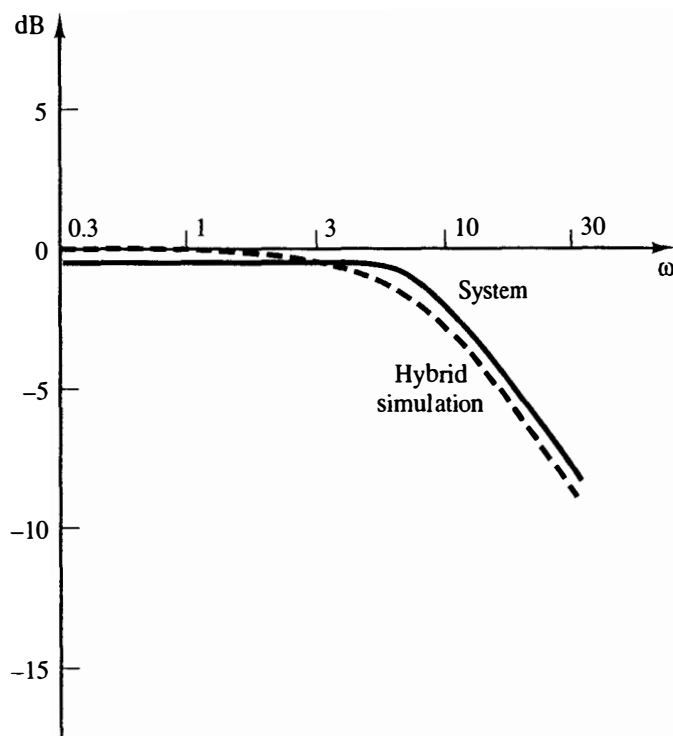


Figure 15-9 Closed-loop frequency response with phase-lead compensation ($T = 5$ ms).

The resulting phase margin was 80° . The system closed-loop frequency response is given in Figure 15-9. Both responses were obtained experimentally. The analog computer simulation, with filter, was linear; thus the closeness of the two responses indicates little nonlinear effects in the system. The system step response is given in Figure 15-10.

A phase-lag filter was also designed, using the procedure given in Chapter 8. The resultant filter transfer function is

$$D(w) = \frac{1.36(1 + w/0.20)}{1 + w/0.044} \quad (15-11)$$

and thus

$$D(z) = \frac{0.3z - 0.2997}{z - 0.99978} \quad (15-12)$$

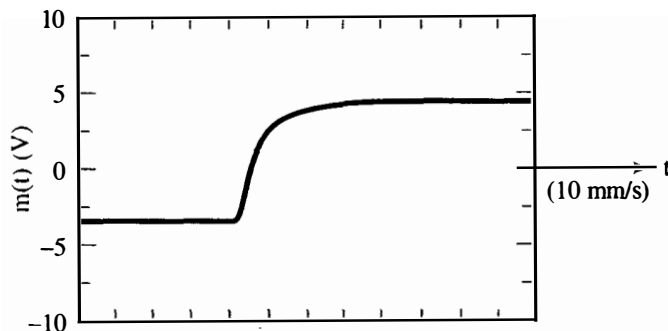


Figure 15-10 Step response of the servomotor system with a phase-lead compensator ($T = 5$ ms).

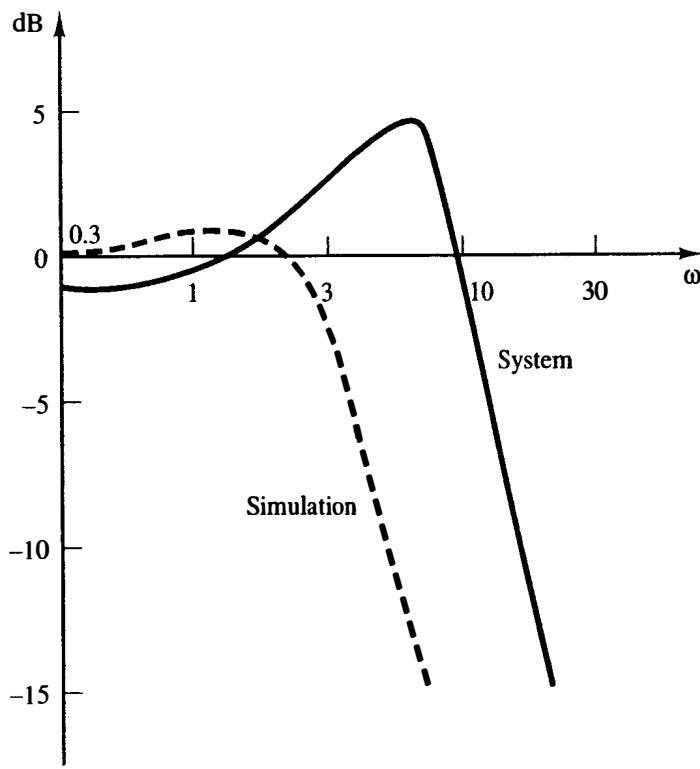


Figure 15-11 Closed-loop frequency response with phase-lag compensation ($T = 5$ ms).

Here the computer calculates

$$y(nT) = 0.3x(nT) - 0.2997x(nT - T) + 0.99978y(nT - T)$$

The phase margin for this system is 60° . However, when the frequency responses were obtained experimentally, the curves in Figure 15-11 resulted. Here the differences in the frequency responses indicate gross nonlinear effects. The system step response is given in Figure 15-12.

15.3 ENVIRONMENTAL CHAMBER CONTROL SYSTEM

This section presents the case study of a digital control system for an environmental chamber designed for the study of plant growth [9]. Two control systems were

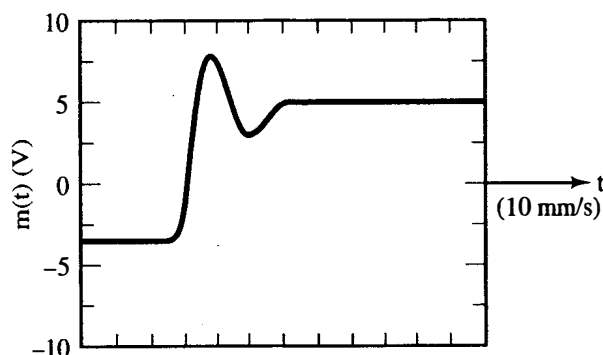


Figure 15-12 Step response of the servomotor system with a phase-lag compensator ($T = 5$ ms).

implemented: one to control dry-bulb temperature and the other to control the carbon dioxide (CO_2) content in the chamber atmosphere. The chamber is constructed of plexiglass with measurements of approximately 0.7 by 1.3 by 2 m. The chamber system is hermetically sealed such that both moisture content and carbon dioxide content can be accurately monitored.

Figure 15-13 gives a hardware description of the temperature control system. The chamber is cooled (air conditioned) to the extent that heaters are required to maintain the desired temperature. Thus the heaters are the controlling elements in the closed-loop system. The resistance bridge is dictated by the temperature sensor used, and has an output in the millivolt range. An operational amplifier increases the amplitude of this signal to the $\pm 5\text{-V}$ range required by the A/D converter. The measured temperature is then subtracted from the desired temperature, which is stored in the TMS 9900 microcomputer system. Next this error signal is processed by the TMS 9900 (the system compensation), resulting in an output signal to the heater interface. The heater interface is a complex logic circuit [9] which converts the computer output signal into the triacs' [10] control pulses. The triacs control the electrical energy into the heaters by in effect controlling the rms voltage applied to the heaters.

The hardware description of the CO_2 control system is given in Figure 15-14. The CO_2 content of the chamber atmosphere is measured by the gas analyzer in parts per million (ppm). This signal is compared to the desired set point, and if the error is negative (a CO_2 deficit), the computer opens a solenoid valve from a CO_2 supply for a length of time dependent on the error magnitude. The control system has no capability to remove excess CO_2 . Because of the time lag required for the gas analysis, the sampling rate for this control system cannot be greater than 0.0222 Hz ($T = 45\text{ s}$). In addition, no compensation is employed. However, as will be shown later, satisfactory control occurs.

The data (number) format for data internal to the computer is given in Figure 15-15 and is a two's-complement fixed-point format [3]. For both control systems, the 16-bit accuracy was found to be sufficient, and convenient.

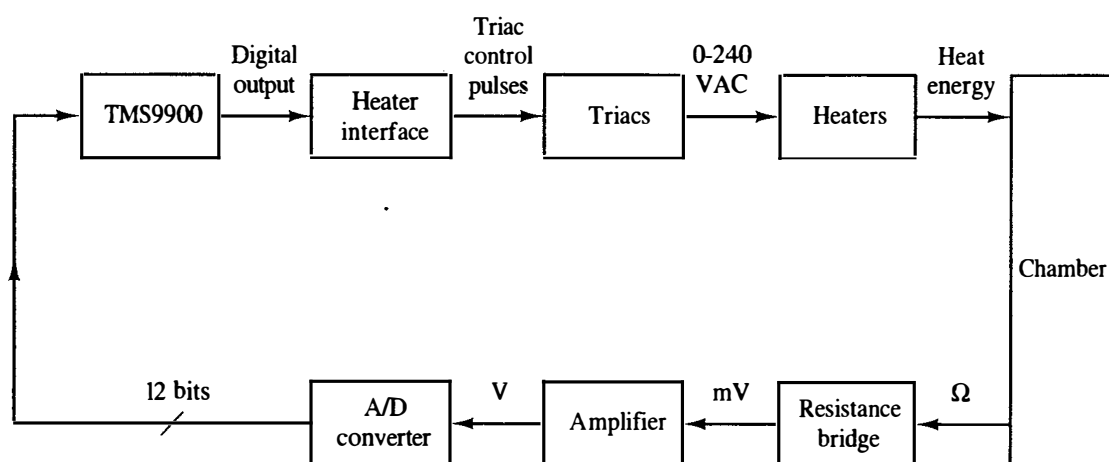
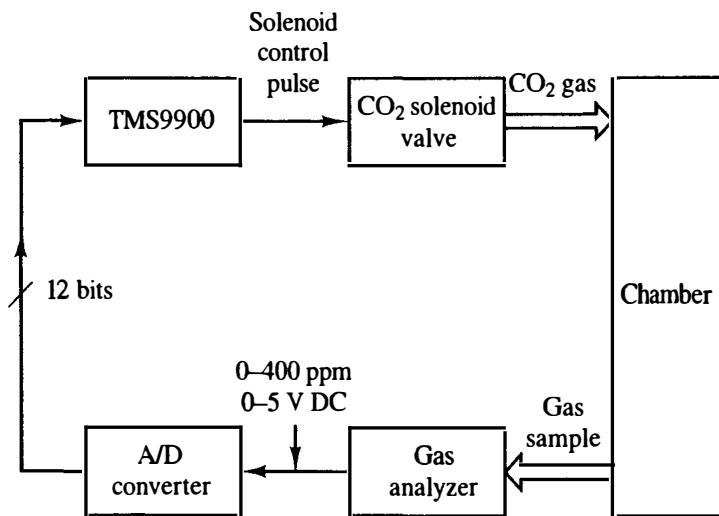


Figure 15-13 Chamber temperature control hardware diagram.

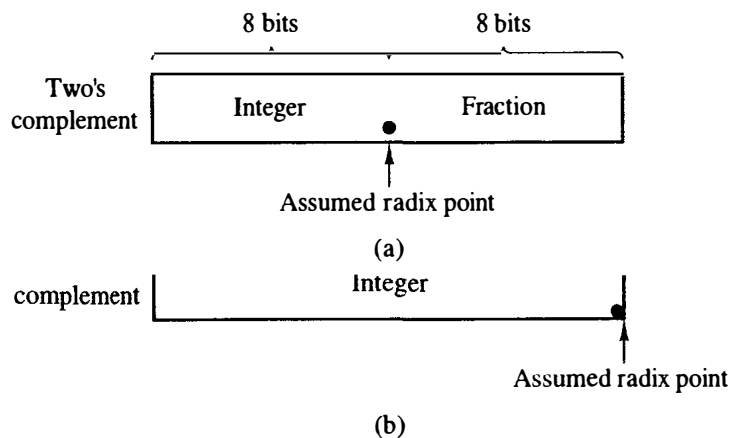

 Figure 15-14 CO₂ control hardware.

Temperature Control System

The temperature control system was designed empirically. Because of the complexity of the system, no attempt was made to derive a plant model. Instead, a step response was run between the heater input and the temperature sensor output. The temperature rise was found to be approximately exponential with a time constant of 60 s. Thus as a first try, the sample period was set to 6 s.

To aid in the empirical design, a proportional-plus-integral-plus-derivative (PID) controller was chosen for implementation (see Chapter 8). The analog version of the PID filter is

$$m(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt} \quad (15-13)$$


 Figure 15-15 (a) Binary format for temperature software; (b) binary format for CO₂ software.

where $e(t)$ is the controller input and $m(t)$ is the controller output. The discrete controller implementation of the integrator was chosen to be

$$m_i(k) = \frac{T}{2}[e(k) + e(k-1)] + m_i(k-1) \quad (15-14)$$

and for the differentiator,

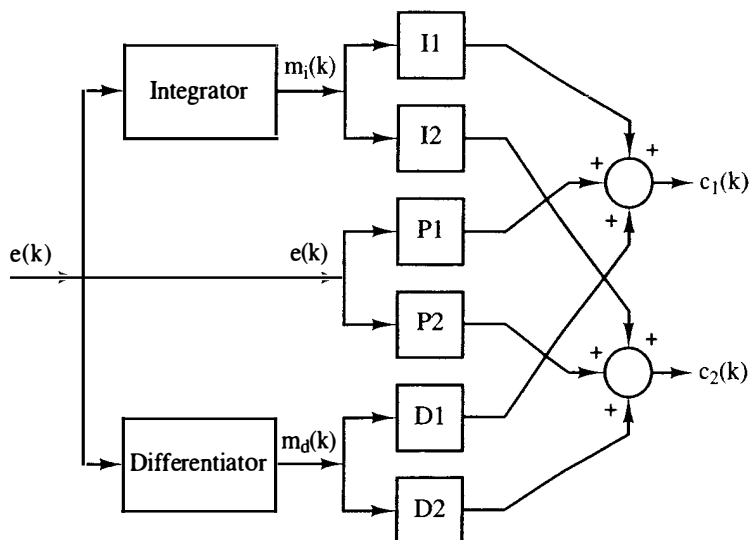
$$m_d(k) = \frac{1}{T}[e(k) - e(k-1)] \quad (15-15)$$

[see (8-34) and (8-37)]. Hence the discrete controller implementation of (15-13) is

$$m(k) = K_P e(k) + K_I m_i(k) + K_D m_d(k) \quad (15-16)$$

It is evident from these equations that four parameters are to be determined: T , K_P , K_I , and K_D . However, if an appropriate choice is made for T such that accurate differentiation and integration occur, the parameters K_P , K_I , and K_D will be independent of T .

The hardware for the temperature control system was chosen prior to design of the controller. In many applications the control system designer will find pre-defined hardware constraints, and must learn to adapt his design techniques, from ideal to practical. Two heaters were specified: one with a slow response (large mass) and one with a faster response (smaller mass). Thus the slow-response heater adds phase lag to the system, which is undesirable. Subsequently, experimentation on the physical system resulted in a choice for the form of the controller as shown in Figure 15-16. Different PID gains (P1, I1, D1, equivalent to K_P , K_I , K_D , respectively) are implemented for the fast-response heater [controlled by the signal $c_1(k)$] from those employed with the slow-response heater, which is controlled by $c_2(k)$. This



where $e(k)$ = error at $t = kT$

$$m_i(k) = T/2 * e/(k) + T/2 * e(k-1) + m_i(k-1)$$

$$m_d(k) = 1/T * e/(k) - 1/T * e(k-1)$$

Figure 15-16 Block diagram of PID filter implementation.

form for the controller results in a better system response than that obtained by controlling both heaters with a single signal.

The controller gains were determined by first implementing a proportional-only controller, and varying $P1$ and $P2$ in Figure 15-16. A typical step response is given in Figure 15-17, with $P1$ and $P2$ both equal to 25. The small oscillation visible in this response is caused by the air conditioner cycling on and off. The control system was commanded to a 10° Celsius step input, from 25°C to 35°C . Note the steady-state error in Figure 15-17. During these tests, the sample period T (T_s in Figure 15-17) was also varied, and a value of $T = 1$ s was chosen; that is, choosing T less than 1 s caused no observable improvement in the system response.

Next the integral term was added to the controller, and a typical step response is given in Figure 15-18. Note the elimination of the steady-state error, as expected (see Section 8.9). Finally, the derivative term was added to the controller, resulting in a typical response as shown in Figure 15-19. Note that no improvement occurs in the rise time for the PID controller, because the heaters are full on for large error signals for both the P controller and the PI controller (a nonlinear effect). Thus the system cannot respond any faster for large errors and this effect is independent of the form of the controller. However, for small errors (e.g., small overshoot), the

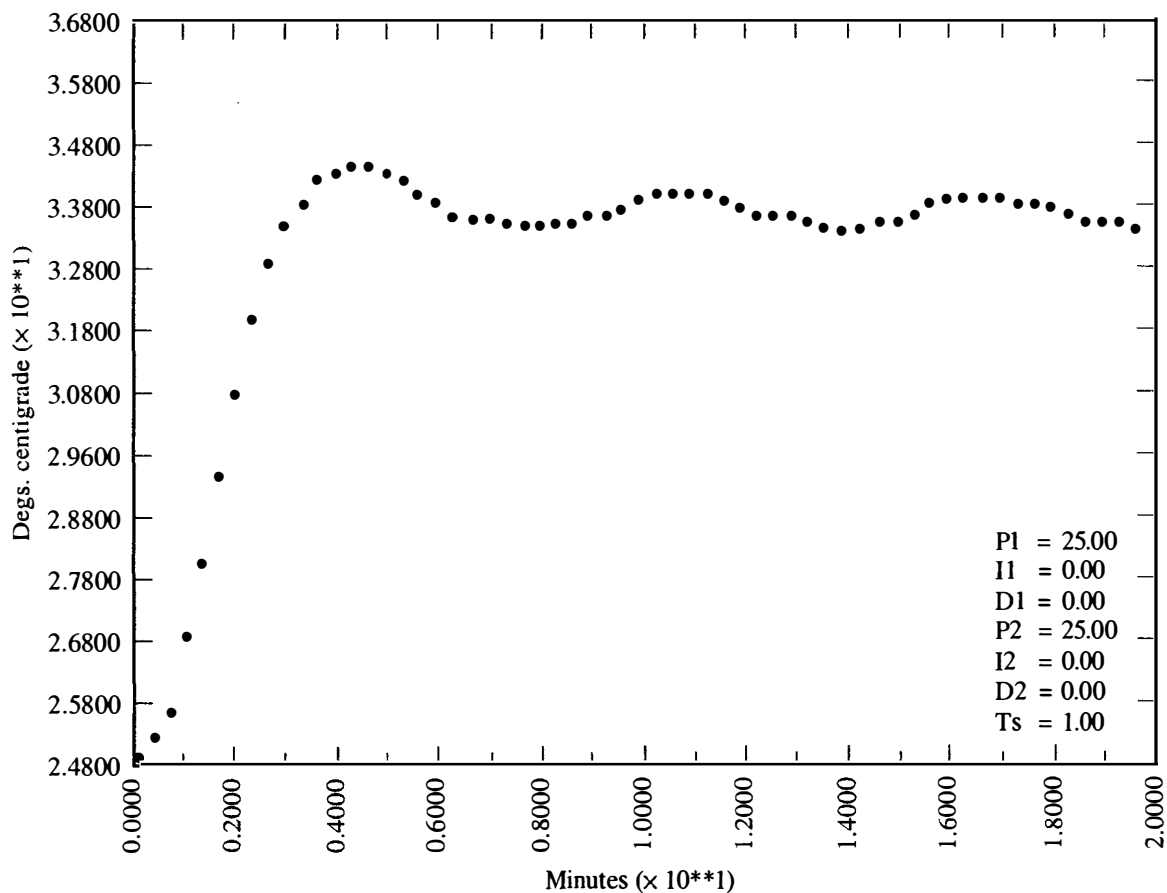


Figure 15-17 Proportional controller step response behavior.

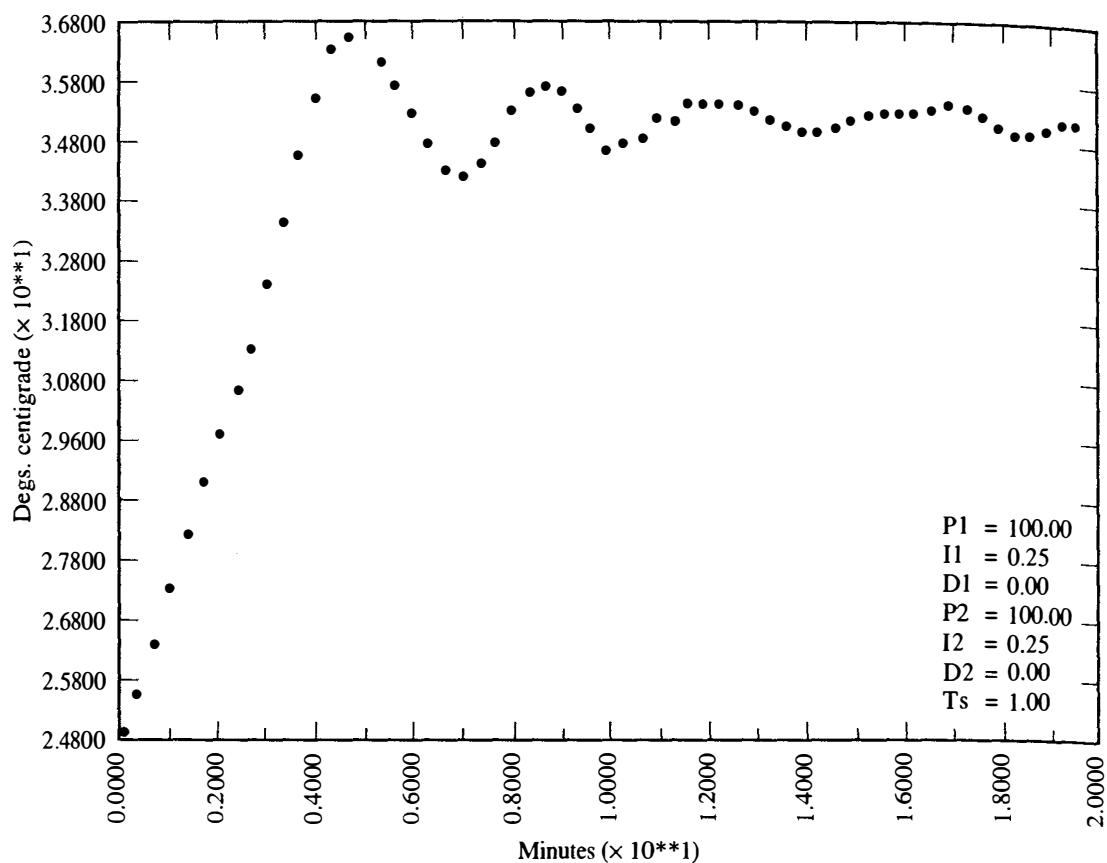


Figure 15-18 PI controller step response behavior.

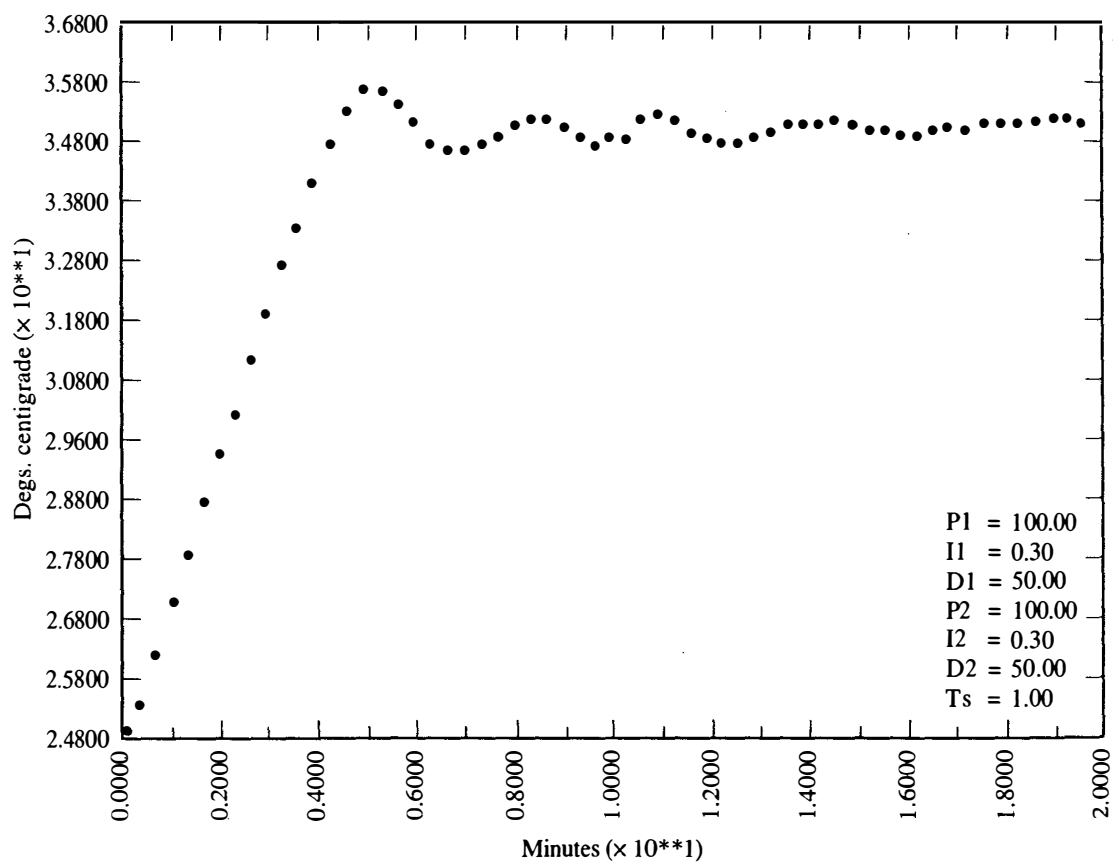


Figure 15-19 PID controller step response behavior.

Once each 250 ms this timer outputs a hardware signal to a specific terminal of the microprocessor, causing the microprocessor to interrupt normal processing and update a real-time clock. The real-time clock is composed of memory locations which contain a numerical description of the present second, minute, hour, and day of the year (Julian day).

A flowchart of the operating software is given in Figure 15-21. Note that the principal program operation is through the interrupts. After system initialization, the computer is in a wait loop except when printing is required. The interrupt system for the TI9900 microprocessor is based on priorities (i.e., certain interrupts take

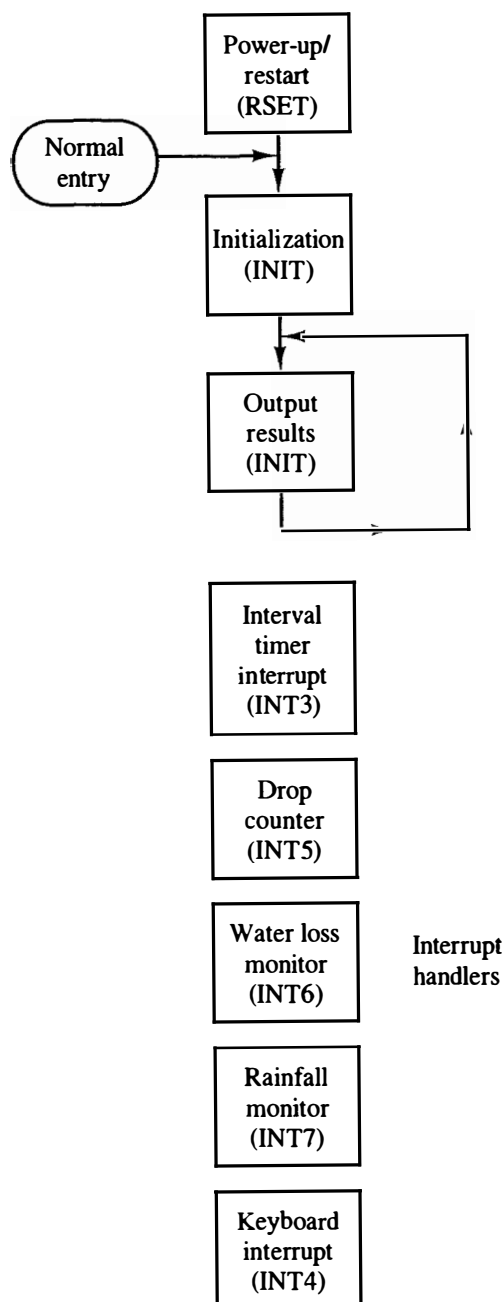


Figure 15-21 Program flowchart.

TABLE 15-1 OPERATING SYSTEM PRIORITIES

Priority ranking	Task description	How implemented
1	Power-up/restart	$\overline{\text{INT0}}$
2	Program initialization	Main program (system not operating)
3	Real-time clock; solenoid valve operation; data output; temperature set point update; control functions	$\overline{\text{INT3}}$
4	Water drop counting	$\overline{\text{INT5}}$
5	Water volume measurement	$\overline{\text{INT6}}$
6	Rainfall measurement	$\overline{\text{INT7}}$
7	Parameter entry	$\overline{\text{INT4}}$
8	Printed results	Main program (system operating)

precedence over others). The interrupt priorities for this system's operating software is given in Table 15-1. The various interrupt service routines will now be described.

INT0. This routine provides an automatic restart of the system following a power failure, and prints a message stating that a power failure has occurred.

INT3. This interrupt is caused by the interval timer decrementing to zero (250 ms has passed), and the service routine is responsible for five tasks: CO₂ value operation, real-time clock service, data output into a data acquisition system, temperature set point update, and sample-interval timing. The sample-interval timing initiates the control processing for the CO₂ controller and the temperature controller at the appropriate sample instants.

INT4, INT5, INT6. These three interrupt service routines allow devices for water measurement to be monitored. Water drop counting and water volume measurement measure the water removed from the chamber atmosphere by the air conditioner. Rainfall measurement records rainfall outside the chamber. All three routines simply count the total number of interrupts occurring (each representing a known volume of water) and reset the devices. The totals are output for data acquisition purposes by the INT3 routine.

15.4 AIRCRAFT LANDING SYSTEM

This section presents the design of an automatic aircraft landing system. The particular control system described is that of the Marine Air Traffic Control and Landing System (MATCALs) [12,13]. The system is illustrated in Figure 15-22. The radar

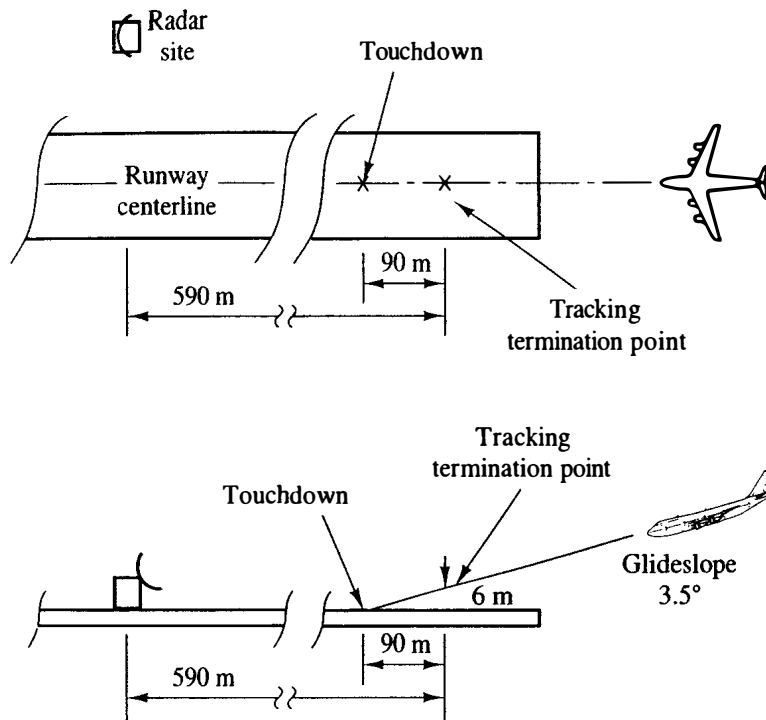


Figure 15-22 Aircraft landing system.

(the control system sensor) employs a phased array antenna, which was chosen because the radar beam can be rapidly repositioned to control a number of planes at sampling rates of up to 40 Hz. However, the positional information provided by the radar system is corrupted with significant noise, which presents problems to the control system designer.

The control operation is composed of two independent (decoupled) control systems: the vertical control system, which keeps the plane on a 3.5° glide slope, and the lateral control system, which maintains the plane on the extended centerline of the runway. These two control systems are illustrated in Figure 15-23. Two separate computer-processing algorithms are utilized in each control system. The first algorithm processes the antenna return signals to determine the aircraft position (centroid position) with respect to the runway touchdown point. The second algorithm comprises the difference equations that describe the digital controllers. The outputs of the controllers are transmitted via a data link to the aircraft and applied to the appropriate autopilot inputs: the pitch autopilot for the vertical control system and the bank autopilot for the lateral control system.

In the following development, only the lateral control system will be discussed. The design of the vertical control system follows similar paths of development.

Plant Model

A block diagram of the lateral control system is given in Figure 15-24. The lateral aircraft dynamics, described by the transfer function $G_L(s)$, include the bank autopilot and typically have a frequency response as shown in Figure 15-25. These dynamics are of course aircraft dependent; the frequency response of the McDonnell

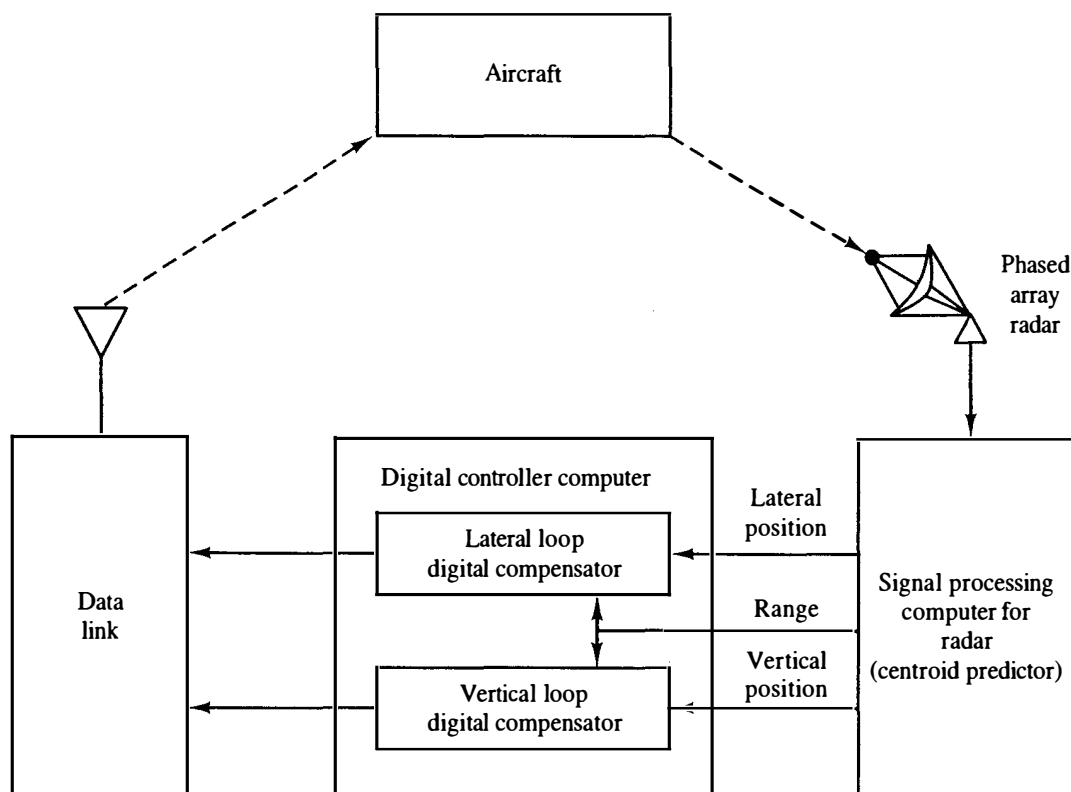


Figure 15-23 MATCALS automatic landing control loops.

Aircraft Company F4J aircraft is given in Figure 15-26, and was obtained from measurements on the aircraft in flight [14]. The lateral aircraft equations of motion will not be developed here (see Ref. 15), but are ninth order for the F4J aircraft with autopilot included. Note from the -40 -dB slope on the magnitude curve of $G_L(j\omega)$ in Figure 15-26 that $G_L(s)$ has a second-order pole at the origin.

A sketch of the Nyquist diagram for the F4J aircraft lateral control system is

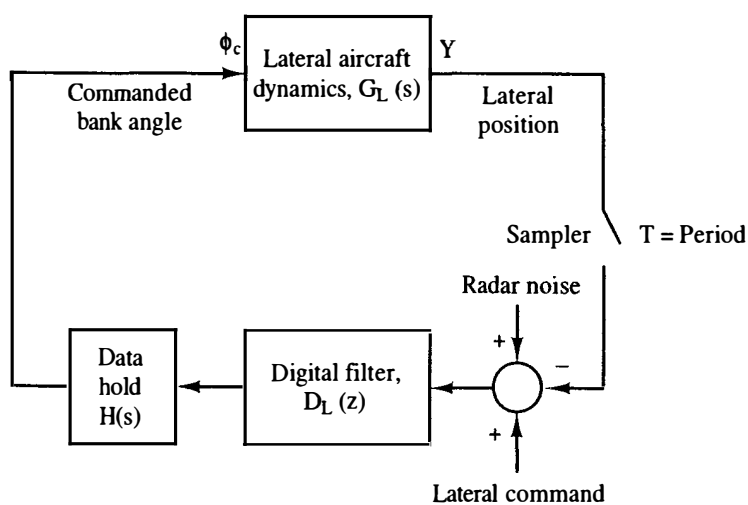


Figure 15-24 Lateral control loop.

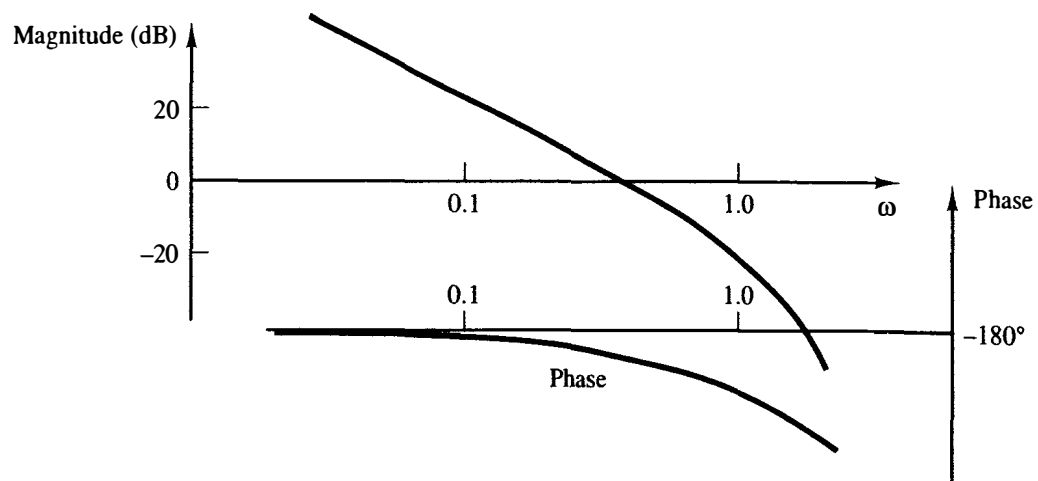


Figure 15-25 Typical frequency response of $G_L(s)$.

given in Figure 15-27, as can be seen from Figure 15-26. Since the aircraft with autopilot must be (open-loop) stable, it is evident that phase-lead compensation must be employed to eliminate the encirclement of the -1 point, in order to stabilize the system.

Design

There are three significant disturbance sources in the lateral control system which must be considered in the system design. The first of these is the radar noise, and is indicated in Figure 15-24. The other two disturbances, which will be considered first, are direct inputs to the aircraft lateral dynamics. Thus each of these disturbances may be modeled as shown in Figure 15-28. Hence the influence of the disturbance on the lateral position $y(k)$ is given by

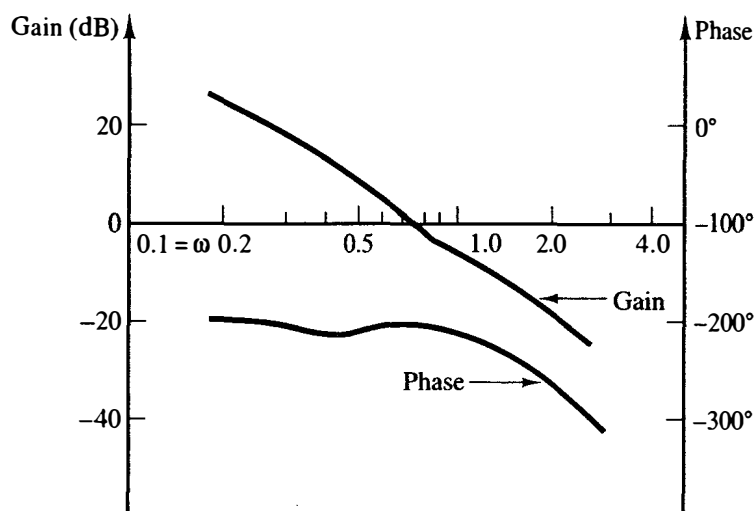


Figure 15-26 F4J lateral frequency response (Y/ϕ_c).

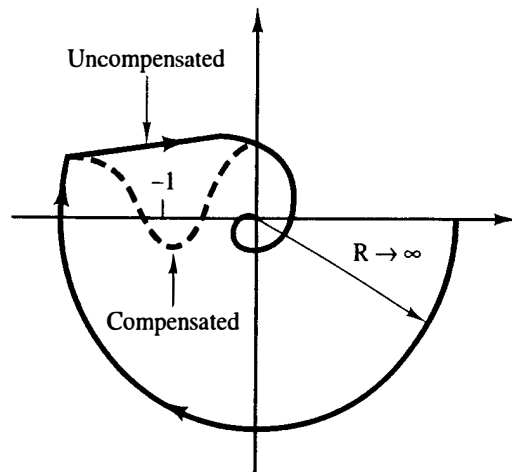


Figure 15-27 Nyquist diagram of $G_L(s)$, and $G_L(s)$ with possible compensation.

$$Y(z) = \frac{\overline{G_D U(z)}}{1 + D(z)G(z)} \quad (15-17)$$

where

$$G(z) = \mathcal{Z} \left[\frac{1 - e^{-Ts}}{s} G_L(s) \right] \quad (15-18)$$

and, as stated previously, $G_L(s)$ has two poles at $s = 0$, resulting in two poles of $G(z)$ at $z = 1$.

One of the disturbances to be modeled as $U(s)$ in (15-17) is a relatively constant error (called a dc bias) in the output of a rate gyroscope in the autopilot. This error can be treated as a system input. Note that, in general, a constant input for the disturbance in Figure 15-28 will result in a constant output $y(k)$ in the steady state. The aircraft would then land to one side of the runway centerline, which is unacceptable. Thus a system requirement is that the final-value theorem when applied to (15-17) yield a value of zero; that is,

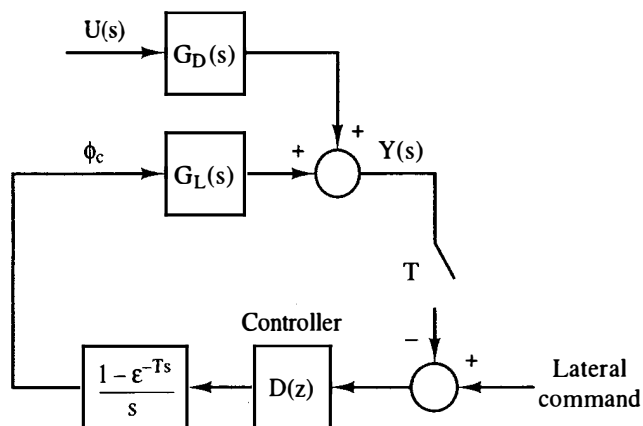


Figure 15-28 Lateral control system with a disturbance.

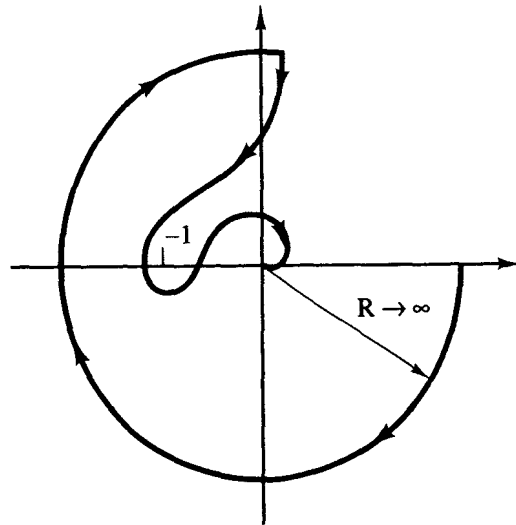


Figure 15-29 Lateral control system Nyquist diagram.

$$\lim_{k \rightarrow \infty} y(k) = \frac{(z-1)\overline{G_D U(z)}}{1 + D(z)G(z)} \Big|_{z=1} = 0 \quad (15-19)$$

with $U(s) = 1/s$. Now both $G_D(s)$ and $G_L(s)$ of Figure 15-28 have two poles at $s = 0$. Thus $\overline{G_D U(z)}$ has three poles at $z = 1$, and $G(z)$ has two poles at $z = 1$. If $D(z)$ is given a pole at $z = 1$, (15-19) is satisfied. Hence a proportional-plus-integral (PI) controller is a system requirement. As was seen from Figure 15-27, phase lead is required for stability. Since the derivative term in the PID controller contributes phase lead [see (8-47)], the PID controller was chosen to compensate this system. The compensated-system Nyquist diagram then appears as shown in Figure 15-29.

The second disturbance source of the type illustrated in Figure 15-28 is the wind. It is desirable that the aircraft not respond to wind inputs, but of course this is not possible. So the controller is designed to reduce the effects of wind. Consider again the effect of the wind disturbance as modeled in Figure 15-28 and equation

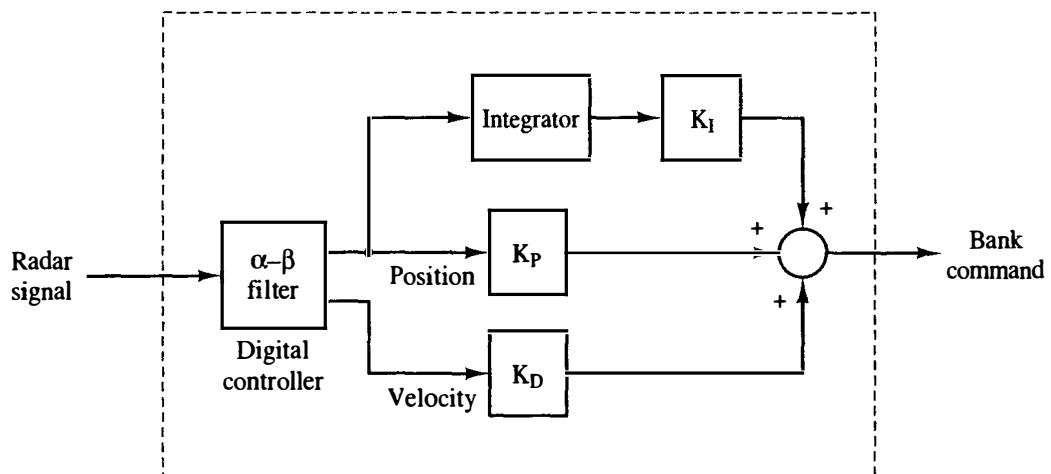


Figure 15-30 Basic form of the controller.

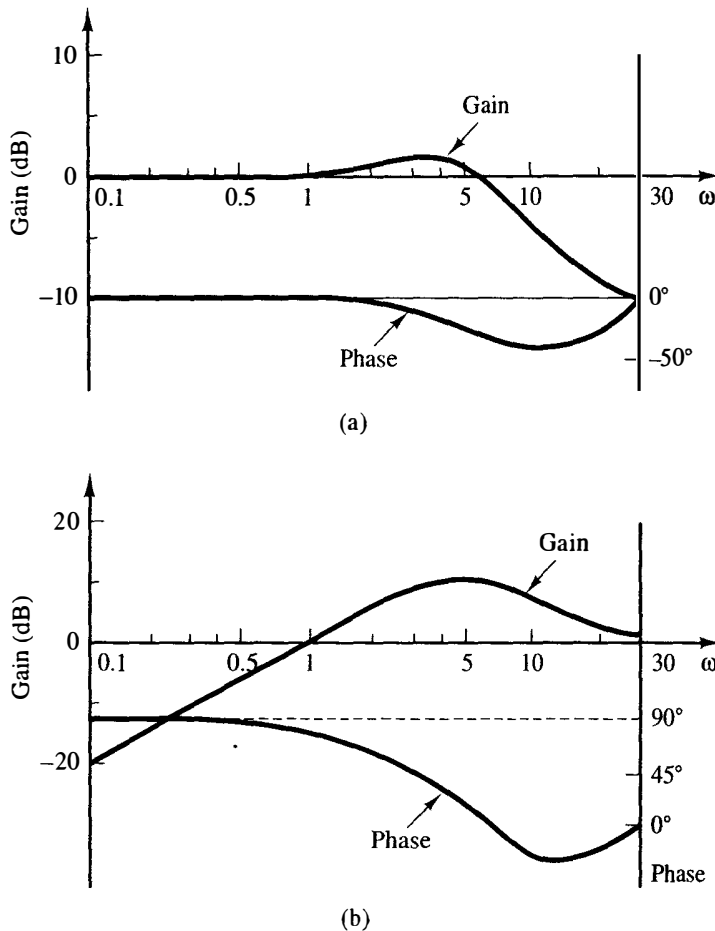
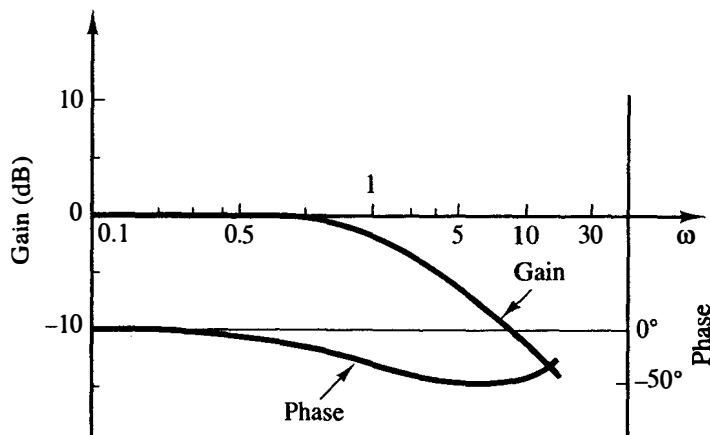


Figure 15-31 (a) α - β position filter response; (b) α - β filter response.

(15-17). Of course, $G_D(s)$ is different for wind disturbances from that for the dc bias disturbance. In (15-17), the response is reduced if $D(z)G(z)$ is made large in the frequency range of the wind (turbulence) input signal. Since the frequency response of $G(z)$ is fixed, $D(z)$ should be designed to increase the system bandwidth. As we have seen in Chapter 8, the derivative path of the PID controller not only introduces phase lead, but also increases system bandwidth. Hence we require a high gain in the derivative path.

The third disturbance source, which is the noise present in the radar system output signal, is modeled as shown in Figure 15-24. Note that the radar noise enters the system at the same point as does the lateral position command signal. Thus a system design that yields a good response to the command input also yields a good response to the radar noise. This noise response is a major problem in the design of this system. For aircraft carrier-based systems [16], a parabolic radar antenna system is used. The signal from this type of antenna system is relatively noise-free, and the noise problem is not as critical as in the land-based system.

α - β filter. To reduce the effects of the radar noise, an α - β tracking filter [17] was chosen to filter the radar signal at the controller input. (The α - β filter equations

Figure 15-32 α -filter frequency response.

are given in Problem 2-21.) This filter is designed to estimate $y(k)$, the aircraft lateral position, and $\dot{y}(k)$, the aircraft lateral velocity, given a noisy radar system output signal. The estimate $y(k)$ is then transmitted to the position path and the integrator path, and the estimate $\dot{y}(k)$ to the derivative path, of the PID compensator. Thus the PID compensator has the basic configuration shown in Figure 15-30.

The frequency responses plotted versus real frequency ω of an α - β filter are given in Figure 15-31. The sample period is $T = 0.1$ s; thus $\omega_s/2$ is 31.4 rad/s. Note that at low frequencies the position filter (Figure 15-31a) has unity gain and no phase shift, but it attenuates high-frequency noise. At low frequencies the velocity filter

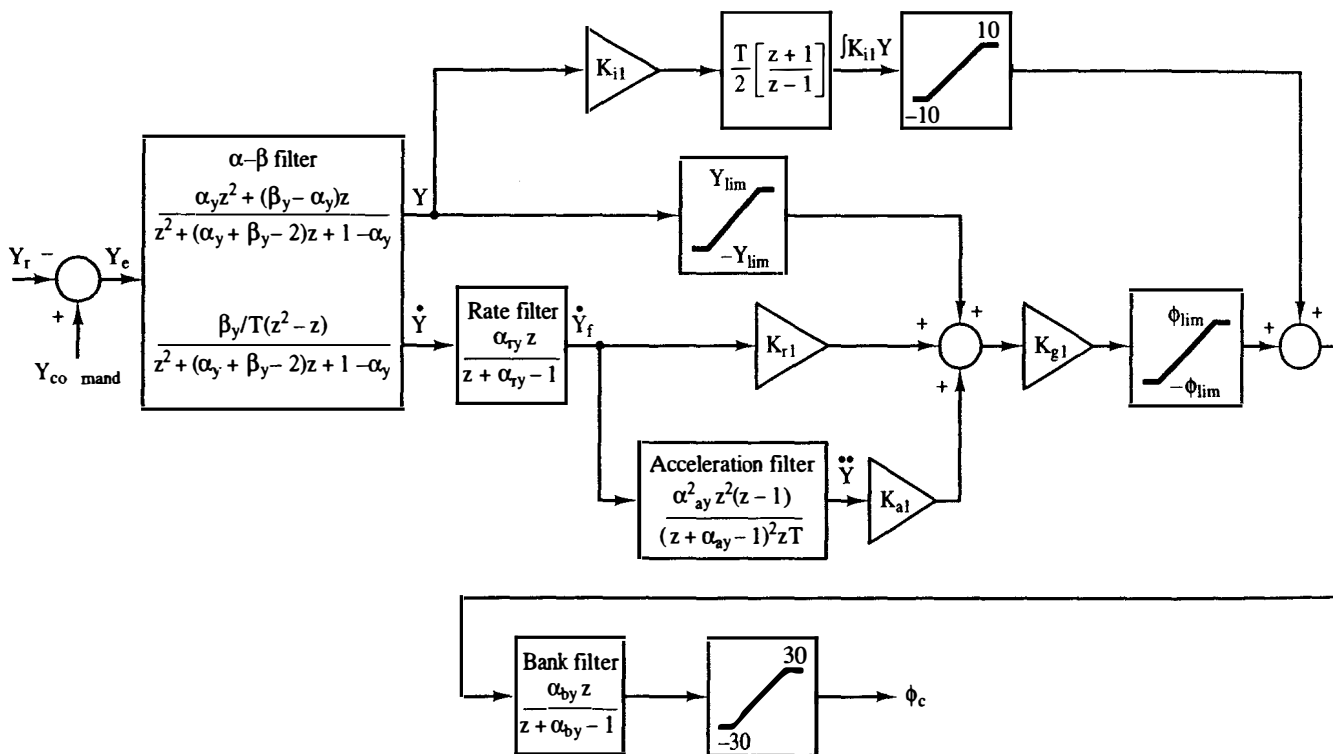


Figure 15-33 Lateral control system controller.

response (Figure 15-31b) has a slope of -20 dB/decade, which is equal to that of an exact differentiator. However, the phase characteristic varies from that of a differentiator, which has a constant phase of 90° . This filter also attenuates high-frequency noise. For the filter plotted, $\alpha = 0.51$ and $\beta = 0.1746$, which are the values used in the F4J controller.

α -filters. The noise-reduction characteristics of the α - β filter do not adequately attenuate the radar noise in this system. To reduce the high-frequency noise further, low-pass filters (at times called α -filters [16]) are added at various points in the PID controller. The α -filter has a transfer function given by

$$D_\alpha(z) = \frac{\alpha z}{z - (1 - \alpha)} \quad (15-20)$$

The frequency response of an α -filter for $\alpha = 0.234$ and $T = 0.1$ s is given in Figure 15-32. This filter is also employed in the F4J controller. As is normal in low-pass filtering, phase lag is added to the system. It is seen from the system Nyquist diagram in Figure 15-29 that phase lag is already a major design problem. Thus the final design must include a trade-off between desired stability margins and radar noise rejection.

The foregoing system requirements resulted in a final filter design illustrated in Figure 15-33. An additional differentiator is added to the $\dot{y}(k)$ path to produce additional phase lead, since that produced by the derivative term was insufficient. The differentiator has the transfer function

$$D_\alpha(z) = \frac{z - 1}{Tz} \quad (15-21)$$

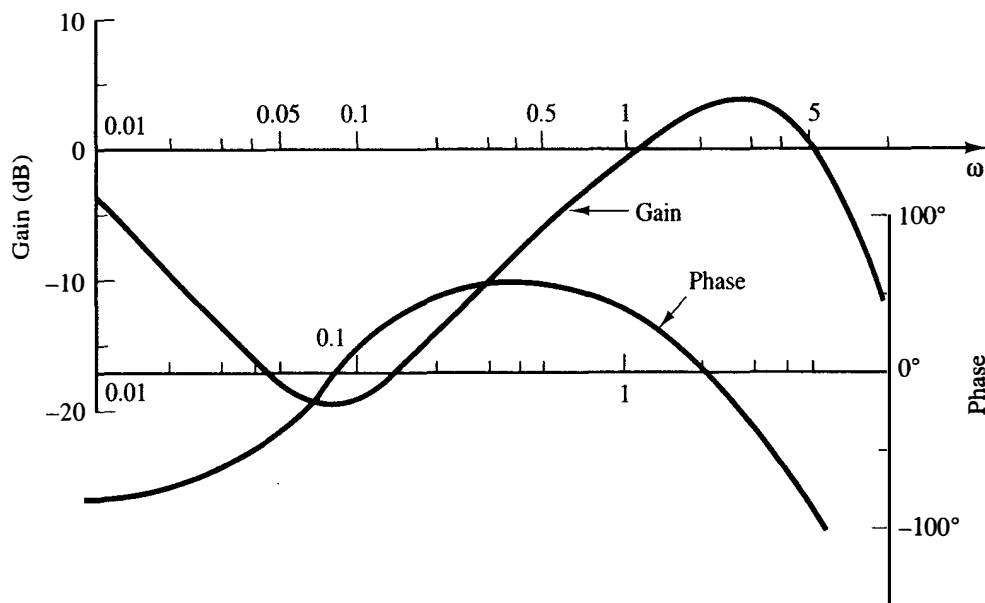


Figure 15-34 F4J lateral controller frequency response, $T = 0.1$ s.

[see (8-38)]. Because of the high-frequency noise amplification from this differentiation of a noisy signal, two sections of α -filters are required for this path. The nonlinearities shown in the filter are not exercised during normal system operation. The limits ϕ_{lim} and Y_{lim} , and the various gains shown, are all functions of range, with the gains increasing as the aircraft nears touchdown. The gains are constant for the final 1500 m of flight. The frequency response (versus real frequency ω) of the lateral controller for the F4J aircraft is given in Figure 15-34; the standard PID filter response is seen, except for the high-frequency attenuation added to reduce noise effects.

The PID path gains used, for range less than 1500 m, are $K_P = 0.1$, $K_I = 0.0033$, $K_D = 0.75$, and (acceleration path) $K_A = 0.75$. The resultant phase margin is 60° , and the gain margin is 8 dB. Two significant nonlinearities in the lateral control system are the mechanical limits on the rotations of the ailerons and of the rudder. For a landing in high wind turbulence, these nonlinearities are exercised and the stability characteristics are degraded.

REFERENCES

1. M. S. Paranjape, "Microprocessor Controller for a Servomotor System," M.S. thesis, Auburn University, Auburn, AL, 1980.
2. ———, *9900 Family Systems Design*. Dallas, TX: Texas Instruments Learning Center, 1978.
3. C. H. Roth, *Fundamentals of Logic Design*, 3d ed. St. Paul, MN: West Publishing Co., 1985.
4. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2d ed. Englewood Cliffs, NJ: Prentice Hall, 1991.
5. J. D. Powell and P. Katz, "Sample Rate Selection for Aircraft Digital Control," *AIAA J.*, Vol. 13, pp. 975-979, Aug. 1975.
6. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, 2d ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 1988.
7. J. C. Hsu and A. U. Meyer, *Modern Control Principles and Applications*. New York: McGraw-Hill Book Company, 1968.
8. A. Gelb and W. E. Vander Velde, *Multiple-Input Describing Functions and Nonlinear Design*. New York: McGraw-Hill Book Company, 1968.
9. ———, *Nonlinear Control Engineering*. New York: McGraw-Hill Book Company, 1968.
10. ———, *Nonlinear Control Engineering*. New York: McGraw-Hill Book Company, 1968.
11. L. A. Leventhal, *Introduction to Microprocessors: Software, Hardware, Programming*. Englewood Cliffs, NJ: Prentice Hall, 1978.
12. "AN/TPN-22 Mode 1 Final Report," Contract N00039-75-C-0021, ITT Gilfillan, Van Nuys, CA, 1979.

13. "Software Implementation ALS Computer Program," Contract N00421-75-C-0058, Bell Aerospace Corporation, Buffalo, NY, Mar. 1975.
14. A. P. Schust, Jr., "Determination of Aircraft Response Characteristics in Approach/Landing Configuration for Microwave Landing System Program," Report FT-61R-73, Naval Air Test Center, Patuxent River, MD, 1973.
15. C. L. Phillips, E. R. Graf, and H. T. Nagle, Jr., "MATCALs Error and Stability Analysis," Report AU-EE-75-2080-1, Auburn University, Auburn, AL, 1975.
16. R. F. Wigginton, "Evaluation of OPS-II Operational Program for the Automatic Carrier Landing System," Naval Electronic Systems Test and Evaluation Facility, Saint Inigoes, MD, 1971.
17. T. R. Benedict and G. W. Bordner, "Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations," *IRE Trans. Autom. Control*, pp. 27-31, July 1962.

APPENDIX I

Design Equations

In this appendix, (8-33) is shown to be solutions to (8-31) and (8-32), by direct substitution. For convenience, $D(j\omega_{w1})$ will be written as D , and similarly for G . Then, from (8-29) and (8-33),

$$\begin{aligned} D &= \frac{a_0 + j\omega_{w1} a_1}{1 + j\omega_{w1} b_1} = \frac{a_0 + j \frac{1 - a_0|G| \cos \theta}{|G| \sin \theta}}{1 + j \frac{\cos \theta - a_0|G|}{\sin \theta}} \\ &= \frac{a_0|G| \sin \theta + j(1 - a_0|G| \cos \theta)}{|G|[\sin \theta + j(\cos \theta - a_0|G|)]} \end{aligned} \quad (\text{A1-1})$$

Then

$$\begin{aligned} |D|^2 &= \frac{a_0^2|G|^2 \sin^2 \theta + 1 - 2a_0|G| \cos \theta + a_0^2|G|^2 \cos^2 \theta}{|G|^2[\sin^2 \theta + \cos^2 \theta - 2a_0|G| \cos \theta + a_0^2|G|^2]} \\ &= \frac{a_0^2|G|^2 + 1 - 2a_0|G| \cos \theta}{|G|^2[a_0^2|G|^2 + 1 - 2a_0|G| \cos \theta]} = \frac{1}{|G|^2} \end{aligned} \quad (\text{A1-2})$$

and (8-31) is satisfied.

To show that the angle of $D(j\omega_{w1})$ is θ , let the numerator angle and the denominator angle in (A1-1) be denoted as θ_1 and θ_2 , respectively. Then, from (A1-1),

$$\begin{aligned}
 \tan(\theta_1 - \theta_2) &= \frac{\tan \theta_1 - \tan \theta_2}{1 + \tan \theta_1 \tan \theta_2} \\
 &= \frac{\frac{1 - a_0|G| \cos \theta}{a_0|G| \sin \theta} - \frac{\cos \theta - a_0|G|}{\sin \theta}}{1 + \frac{\cos \theta - a_0|G| \cos^2 \theta - a_0|G| + a_0^2|G|^2 \cos \theta}{a_0|G| \sin^2 \theta}} \quad (A1-3) \\
 &= \frac{\sin \theta [1 - 2a_0|G| \cos \theta + a_0^2|G|^2]}{\cos \theta [1 - 2a_0|G| \cos \theta + a_0^2|G|^2]} = \tan \theta
 \end{aligned}$$

using

$$\sin^2 \theta - \cos^2 \theta = 1 - 2 \cos^2 \theta$$

Thus the angle of $D(j\omega_{w1})$ is θ , and (8-32) is satisfied.

Mason's Gain Formula

In this appendix we present a technique for finding the transfer function of a system, given either the signal flow graph or the block diagram of that system.

By definition the transfer function of a continuous linear time-invariant (LTI) system is the ratio of the Laplace transform of the output variable to the Laplace transform of the input variable. Let $E(s)$ be the (Laplace transform of the) input variable, $C(s)$ be the output variable, and $G(s)$ be the transfer function. One method of graphically denoting the relationship

$$C(s) = G(s)E(s) \quad (\text{A2-1})$$

is through a *block diagram*, as shown in Figure A2-1a. For the block shown, the output is equal *by definition* to the transfer function given in the block multiplied by the input. The input and the output are defined by the directions of the arrowheads, as shown.

A signal flow graph is also used to denote graphically the transfer function relationship. The signal flow graph that represents (A2-1) is given in Figure A2-1b. Each signal is represented by a *node* in the signal flow graph, as shown by $E(s)$ and $C(s)$ in the figure. Each transfer function is represented by a *branch*, shown in the figure by the line and arrowhead, with the transfer function written near the arrowhead. *By definition*, the signal out of a branch is equal to the transfer function of the branch multiplied by the signal into the branch.

Two very important points are to be made. First, a block diagram and a signal flow graph contain exactly the same information. There is no advantage to one over the other; there is only personal preference. Next, a block diagram (or signal flow

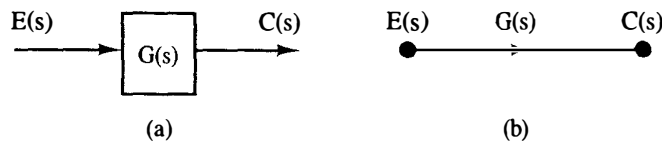


Figure A2-1 Block diagram and signal flow graph elements.

graph) is the graphical representation of an equation or a set of equations, since the block diagram is constructed from the equations.

One additional element is required to represent equations by a block diagram. This element is the *summing junction*, which is illustrated in Figure A2-2 for the equation

$$C(s) = G_1(s)E_1(s) + G_2(s)E_2(s) - G_3(s)E_3(s) \quad (\text{A2-2})$$

For the block diagram, the summing junction is represented by a circle, as in Figure A2-2a. *By definition*, the signal out of the summing junction is equal to the sum of the signals into the junction, with the sign of each component determined by the sign placed next to the arrowhead of the component. Note that whereas a summing junction can have any number of inputs, we show only one output.

For the signal flow graph, the function of the summing junction is inherently implemented by a node. A summing junction is represented by branches into a node, as illustrated in Figure A2-2b. *By definition*, the signal at a node is equal to the sum of the signals from the branches connected into that node.

We now present a procedure that allows us to find the transfer function, by inspection, of either a block diagram or a signal flow graph. This procedure is called *Mason's gain formula* [1]. However, even though this procedure is relatively simple, it must be used with extreme care, since terms in either the numerator or the denominator of the transfer function can *easily* be overlooked. Furthermore, there is no method available that will give an indication in the case that terms have been overlooked. However, Mason's gain formula can be applied to simple systems with some confidence, after one gains experience.

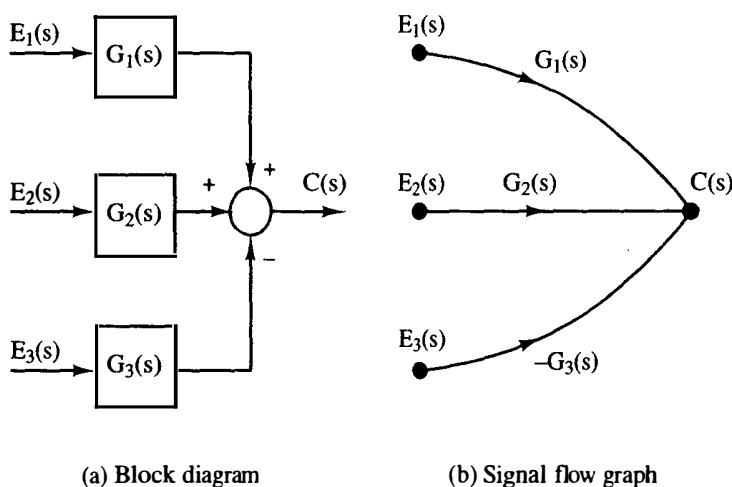


Figure A2-2 Equivalent examples.

In this section we give the rules for Mason's gain formula as applied to signal flow graphs. Exactly the same rules apply to block diagrams. First, definitions are given

Source node A *source node* is a node for which signals flow *only away* from the node; that is, for the branches connected to a source node, the arrowheads are all directed away from the node.

Sink node A *sink node* is a node for which signals flow *only towards* the node.

Path A *path* is a continuous connection of branches from one node to another with all arrowheads in the same direction; that is, all signals flow in the same direction from the first node to the second one.

Loop A *loop* is a closed path (with all arrowheads in the same direction) in which no node is encountered more than once. Note that a source node cannot be a part of a loop, since each node in the loop must have at least one branch into the node and at least one branch out.

Forward path A *forward path* is a path that connects a source node to a sink node, in which no node is encountered more than once.

Path gain The *path gain* is the product of the transfer functions of all branches that form the path.

Loop gain The *loop gain* is the product of the transfer functions of all branches that form the loop.

Nontouching Two loops are *nontouching* if these loops have no nodes in common. A loop and a path are nontouching if they have no nodes in common.

All these definitions are illustrated using the signal flow graph of Figure A2-3. There are two loops in the flow graph, one with a gain of $-G_2 H_1$ and the other with a gain of $-G_4 H_2$. Note that these two loops do not touch. In addition, there are two forward paths connecting the input $R(s)$ and the output $C(s)$. One of these forward paths has a gain of $G_1 G_2 G_3 G_4 G_5$, and the other has a gain of $G_6 G_4 G_5$. Note that the forward path $G_6 G_4 G_5$ does not touch the loop $-G_2 H_1$ but does touch the other loop. The path $G_1 G_2 G_3 G_4 G_5$ touches both loops.

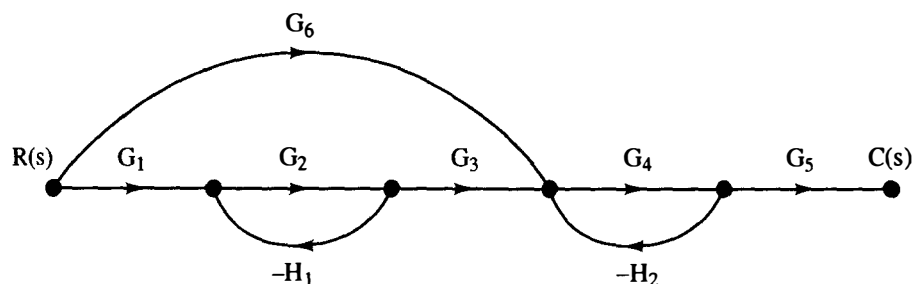


Figure A2-3 Signal flow graph.

Based on the preceding definitions, we may now state Mason's gain formula. The formula gives the transfer function from a source (input) node to a sink (output) node *only* and may be stated as

$$T = \frac{1}{\Delta} \sum_{k=1}^p M_k \Delta_k = \frac{1}{\Delta} (M_1 \Delta_1 + M_2 \Delta_2 + \cdots + M_p \Delta_p) \quad (\text{A2-3})$$

where T is the gain (transfer function) from the input node to the output node, p is the number of forward paths, and

$$\begin{aligned} \Delta = & 1 - (\text{sum of all individual loop gains}) \\ & + (\text{sum of the products of the loop gains of all possible combinations} \\ & \quad \text{of nontouching loops taken two at a time}) \\ & - (\text{sum of the products of the loop gains of all possible combinations} \\ & \quad \text{of nontouching loops taken three at a time}) \\ & + (\text{sum of the products of the loop gains of all possible combinations} \\ & \quad \text{of nontouching loops taken four at a time}) \\ & - (\cdots) \end{aligned}$$

M_k = path sign of the k th forward path

Δ_k = value of Δ for that part of the flow graph not touching the k th forward path

An example is now given to illustrate the use of Mason's gain formula.

Example

Consider again the system of Figure A2-3. Let L_i be the gain of the i th loop. Then the gains of the only two loops can be written as

$$L_1 = -G_2 H_1; \quad L_2 = -G_4 H_2$$

Two forward paths are present in Figure A2-3; the forward path gains can be expressed as

$$M_1 = G_1 G_2 G_3 G_4 G_5; \quad M_2 = G_6 G_4 G_5$$

The value of Δ can be written directly from Figure A2-3:

$$\Delta = 1 - (L_1 + L_2) + L_1 L_2 = 1 + G_2 H_1 + G_4 H_2 + G_2 G_4 H_1 H_2$$

The last term is present in this equation since the two loops do not touch; that is, the loops have no nodes in common.

The determination of the Δ_k of (A2-3) is more difficult. As just stated, the value of Δ_1 is the value of Δ for that part of the flow graph not touching the first forward path. One method of evaluating Δ_1 is to redraw the flow graph with the first forward path removed. Of course, all nodes of the first forward path must also be removed. Figure A2-4a gives the results of removing the first forward path. The second part of the figure illustrates removing the second forward path. Hence Δ_1 is simply the Δ of the flow graph of Figure A2-4a, and Δ_2 is that of Figure A2-4b. Thus we can write

$$\Delta_1 = 1; \quad \Delta_2 = 1 - (-G_2 H_1)$$

since Figure A2-4a has no loops and Figure A2-4b has one loop. Thus from (A2-3) we can write the transfer function of the system of Figure A2-3 as

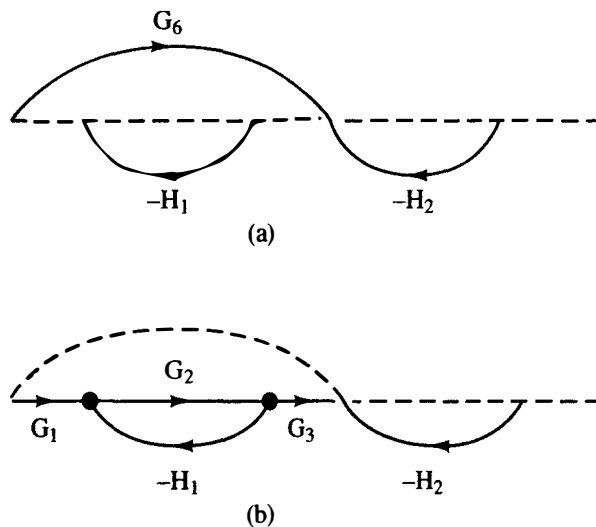


Figure A2-4 Flow graphs with forward path removed.

$$T = \frac{M_1 \Delta_1 + M_2 \Delta_2}{\Delta} = \frac{G_1 G_2 G_3 G_4 G_5 + G_6 G_4 G_5 (1 + G_2 H_1)}{1 + G_2 H_1 + G_4 H_2 + G_2 G_4 H_1 H_2}$$

Note that even for this relatively simple flow graph, many terms appear in the transfer function. Hence terms can easily be overlooked. The only methods available to verify the results of Mason's gain formula are algebraic methods such as Cramer's rule [2].

As stated earlier, Mason's gain formula can be applied directly to block diagrams. In the definitions which appear previously in this appendix, the following replacements are made, with the term *signal* defined as any input or any output of a block or a summing junction.

Signal flow graph	Block diagram
input node	→ input signal
output node	→ output signal
branch	→ block
node	→ signal

Note that we can identify any internal signal as the output signal.

REFERENCES

1. S. J. Mason, "Feedback Theory: Some Properties of Flow Graphs," *Proc. IRE*, Vol. 41, pp. 1144–1156, Sept. 1953.
2. C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2ded. Englewood Cliffs, NJ: Prentice Hall, 1991.

APPENDIX III

Evaluation of $E^*(s)$

$E^*(s)$, as defined by (3-7), has limited usefulness in an analysis because it is expressed as an infinite series. However, for many useful time functions $E^*(s)$ can be expressed in closed form. This closed form of $E^*(s)$ will now be derived.

From (3-6),

$$e^*(t) = e(t)\delta_T(t) \quad (\text{A3-1})$$

However, this function is the inverse Laplace transform of $E^*(s)$ only for cases in which $e(t)$ is continuous at all sampling instants. Problems arise if $e(t)$ is discontinuous at any sampling instant, since the inverse Laplace transform evaluated at a discontinuity will give the average value of the discontinuity. As defined in Section 3.3, however, if $e(t)$ is discontinuous at a sampling instant, then at that instant $E^*(s)$ assumes the value of $e(t)$ from the right. For example, if $e(t)$ is discontinuous at the origin, the calculation of $E^*(s)$ from (A3-1) would yield a value for the function at $t = 0$ of $\frac{1}{2}e(0)$. Thus, if $e(0) \neq 0$, (A3-1) must be expressed as

$$e^*(t) = e(t)\delta_T(t) + e(0)\delta(t) \quad (\text{A3-2})$$

If $e(t)$ is discontinuous at other sampling instants, then impulse functions with values $\Delta e(kT)\delta(t - kT)$ must be added to (A3-2), where $\Delta e(kT)$ is the amplitude of the discontinuity of $e(t)$ at $t = kT$; that is

$$\Delta e(kT) = e(kT^+) - e(kT^-)$$

where $e(kT^-) = e(t)$ evaluated at $t = kT - \epsilon$, and where ϵ is arbitrarily small.

The following derivation applies for the case in which $e(t)$ is continuous at all sampling instants. From (A3-1),

$$E^*(s) = E(s) * \Delta_T(s) \quad (\text{A3-3})$$

where $*$ denotes complex convolution [1] and $\Delta_T(s)$ is the Laplace transform of $\delta_T(t)$. From (3-5),

$$\Delta_T(s) = 1 + e^{-Ts} + e^{-2Ts} + \dots = \frac{1}{1 - e^{-Ts}} \quad (\text{A3-4})$$

Therefore, the poles of $\Delta_T(s)$ occur at values of s for which

$$e^{-Ts} = 1 \quad (\text{A3-5})$$

Equation (A3-5) is satisfied for $s = j(2\pi n/T) = jn\omega_s$, $n = 0, \pm 1, \pm 2, \dots$, where ω_s is the sampling frequency expressed in radians per second. The poles of $\Delta_T(s)$ are shown on the s -plane plot of Figure A3-1.

By definition, equation (A3-3) can be expressed as

$$\begin{aligned} E^*(s) &= \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} E(\lambda) \Delta_T(s - \lambda) d\lambda \\ &= \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} E(\lambda) \frac{1}{1 - e^{-T(s-\lambda)}} d\lambda \end{aligned} \quad (\text{A3-6})$$

where the poles of the integrand of (A3-6) occur as shown in Figure A3-2. The value of c must be chosen such that the poles of $E(\lambda)$ are to the left of the path of integration, and the value of s must be selected so that the poles of $\Delta_T(s - \lambda)$ are to the right of the path of integration [1]. An examination of (A3-6) and Figure A3-2 indicates that $E^*(s)$ can be expressed as

$$E^*(s) = \frac{1}{2\pi j} \oint_{\gamma} E(\lambda) \Delta_T(s - \lambda) d\lambda - \frac{1}{2\pi j} \int E(\lambda) \Delta_T(s - \lambda) d\lambda \quad (\text{A3-7})$$

Consider the second integral of (A3-7). This term is zero if $\lim_{\lambda \rightarrow \infty} \lambda E(\lambda) = 0$ [i.e., if $e(0) = 0$]. If $e(0)$ is not zero, the second integral can be shown [2] to be equal to $e(0)/2$. However, from (A3-2), if $e(0)$ is not zero, an additional term equal to $e(0)/2$ must be added to (A3-7). Therefore, in either case,

$$E^*(s) = \frac{1}{2\pi j} \oint_{\gamma} E(\lambda) \Delta_T(s - \lambda) d\lambda \quad (\text{A3-8})$$

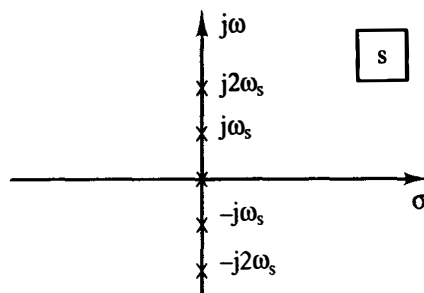


Figure A3-1 Poles of $\Delta_T(s)$.

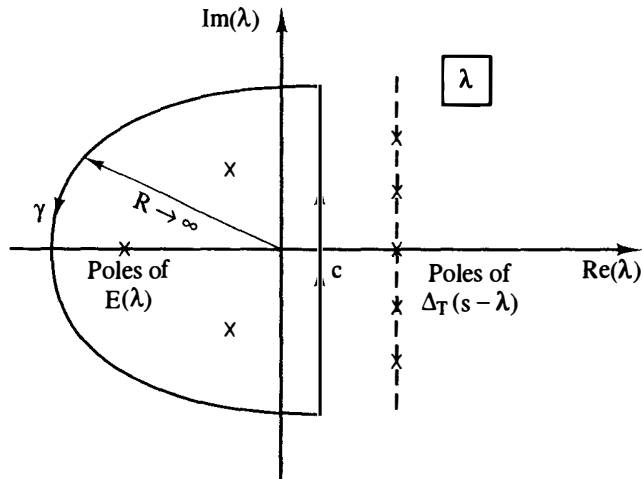


Figure A3-2 Pole locations for the integrand of (A3-6).

The reader interested in a more detailed mathematical development of (A3-8) is referred to Refs. 1 to 3.

The theorem of residues [3] can be used to evaluate (A3-8).

Theorem of Residues. If C is a closed curve and if $f(z)$ is analytic within and on C except at finite number of singular points in the interior of C , then

$$\oint_C f(z) dz = 2\pi j[r_1 + r_2 + \cdots + r_n]$$

where r_1, r_2, \dots, r_n are the residues of $f(z)$ at the singular points within C .

Using this theorem, we can express (A3-8) as

$$E^*(s) = \sum_{\text{at poles of } E(\lambda)} \left[\text{residues of } E(\lambda) \frac{1}{1 - e^{-T(s-\lambda)}} \right] \quad (\text{A3-9})$$

Recall that the residues are evaluated at the poles of the function $E(\lambda)$, since as shown in Figure A3-2, the singular points that lie within the closed contour are derived from this function. For the case in which $E(s)$ has only simple poles, we may use (2-33). Or, by letting

$$E(s) = \frac{N(s)}{D(s)} \quad (\text{A3-10})$$

where $N(s)$ and $D(s)$ are polynomials in s , (A3-9) can be expressed as [4]

$$E^*(s) = \sum_n \frac{N(\lambda_n)}{D'(\lambda_n)} \frac{1}{1 - e^{-T(s-\lambda_n)}} \quad (\text{A3-11})$$

where λ_n are the locations of simple poles of $E(\lambda)$ and

$$D'(\lambda) = \frac{dD(\lambda)}{d\lambda}$$

For multiple-ordered poles, the residue may be found using the expression illustrated in (2-34).

It is of interest to consider the case in which the function $e(t)$ contains a time delay. For example, consider a delayed signal of the type

$$e(t) = e_1(t - t_0)u(t - t_0) \quad (\text{A3-12})$$

Then

$$E(s) = \epsilon^{-t_0 s} \mathcal{L}[e_1(t)] = \epsilon^{-t_0 s} E_1(s) \quad (\text{A3-13})$$

For this case, in general $\lim_{\lambda \rightarrow \infty} \lambda E(\lambda)$ is not finite in the second integral in (A3-7) (see Figure A3-2), and thus (A3-9) does not apply. Special techniques are required to find the starred transform of a delayed signal in closed form, and these techniques are presented in Chapter 4, where the modified z-transform is developed. However, for the special case in which the time signal is delayed a whole number of sampling periods, (A3-9) can be applied in a slightly different form,

$$[\epsilon^{-kTs} E_1(s)]^* = \epsilon^{-kTs} \sum_{\substack{\text{at poles} \\ \text{of } E_1(\lambda)}} \left[\text{residues of } E_1(\lambda) \frac{1}{1 - \epsilon^{-T(s-\lambda)}} \right] \quad (\text{A3-14})$$

where k is a positive integer.

Equation (A3-6) can also be evaluated using the path α shown in Figure A3-3. In this case, the poles of $\Delta_T(s - \lambda)$ are enclosed by α . For the case in which $\lim_{\lambda \rightarrow \infty} \lambda E(\lambda)$ is zero, the integral around the infinite semicircular portion of α is zero and

$$E^*(s) = - \sum_{\substack{\text{poles of} \\ \Delta_T(s-\lambda)}} [\text{residues of } E(\lambda) \Delta_T(s - \lambda)] \quad (\text{A3-15})$$

This relationship is derived using the same steps that were used in the derivation of (A3-9). Recall that

$$\Delta_T(s - \lambda) = \frac{1}{1 - \epsilon^{-T(s-\lambda)}} = \frac{N(\lambda)}{D(\lambda)} \quad (\text{A3-16})$$

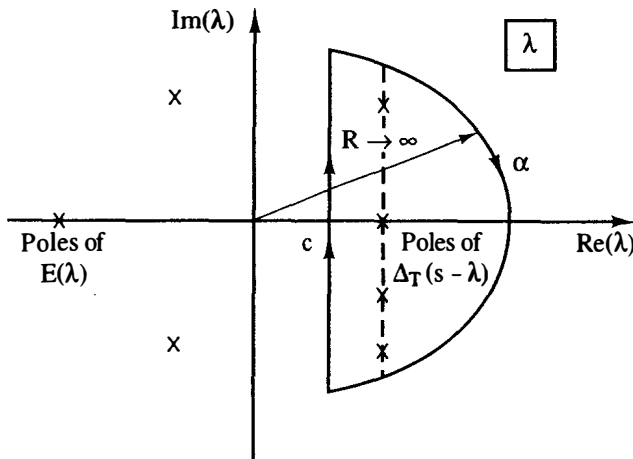


Figure A3-3 Integration path in the λ -plane.

has simple poles located at periodic intervals along a fixed line in the λ -plane as shown in Figure A3-3, and $E^*(s)$ can be expressed as

$$E^*(s) = - \sum_{n=-\infty}^{\infty} \frac{N(\lambda_n)}{D'(\lambda_n)} E(\lambda_n) \quad (\text{A3-17})$$

The poles of $\Delta_T(s - \lambda)$ occur at

$$s - \lambda_n = j \frac{2\pi n}{T} = jn\omega_s, \quad n = 0, \pm 1, \pm 2, \dots \quad (\text{A3-18})$$

or, solving for λ_n ,

$$\lambda_n = s - jn\omega_s, \quad n = 0, \pm 1, \pm 2, \dots \quad (\text{A3-19})$$

Now in this case

$$D'(\lambda) = -T\epsilon^{-T(s-\lambda)} \quad (\text{A3-20})$$

Thus

$$D'(\lambda_n) = -T\epsilon^{-j2\pi n} = -T \quad (\text{A3-21})$$

Then using (A3-16) and (A3-21), (A3-17) can be expressed as

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) \quad (\text{A3-22})$$

For the case in which $\lim_{\lambda \rightarrow \infty} E(\lambda) = 0$, but $e(0) \neq 0$, the integral around the infinite semicircular portion of α is also zero. However, because of the additional term in (A3-2), (A3-22) becomes

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) + \frac{e(0^+)}{2} \quad (\text{A3-23})$$

Therefore, the general expression for (A3-23) is

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) + \frac{1}{2} \sum_{n=0}^{\infty} \Delta e(nT) \epsilon^{-nTs} \quad (\text{A3-24})$$

where $\Delta e(nT)$ is the amplitude of the discontinuity of $e(t)$ at $t = nT$.

In summary, there are three expressions of $E^*(s)$. These are:

$$E^*(s) = \sum_{n=0}^{\infty} e(nT) \epsilon^{-nTs} \quad (\text{A3-25})$$

$$E^*(s) = \sum_{\substack{\text{at poles} \\ \text{of } E(\lambda)}} \left[\text{residues of } E(\lambda) \frac{1}{1 - \epsilon^{-T(s-\lambda)}} \right] \quad (\text{A3-26})$$

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) + \frac{1}{2} \sum_{n=0}^{\infty} \Delta e(nT) \epsilon^{-nTs} \quad (\text{A3-27})$$

Equation (A3-25) is the defining equation for $E^*(s)$, and (A3-26) and (A3-27) are derived from (A3-25).

REFERENCES

1. M. F. Gardner and J. L. Barnes, *Transients in Linear Systems*, Vol. I. New York: John Wiley & Sons, Inc., 1942.
2. C. L. Phillips, D. L. Chenoweth, and R. K. Cavin III, "z-Transform Analysis of Sampled-Data Control Systems without Reference to Impulse Functions," *IEEE Trans. Educ.*, Vol. E-11, pp. 141-144, June 1968.
3. C. R. Wylie, Jr., *Advanced Engineering Mathematics*, 4th ed. New York: McGraw-Hill Book Company, 1975.
4. E. A. Guillemin, *The Mathematics of Circuit Analysis*. New York: John Wiley & Sons, Inc., 1949.

APPENDIX IV

Review of Matrices

Presented in this appendix is a brief review of matrices. Those readers interested in more depth are referred to Refs. 1 to 5.

The study of matrices originated in linear equations. As an example, consider the equations

$$\begin{aligned}x_1 + x_2 + x_3 &= 3 \\x_1 + x_2 - x_3 &= 1 \\2x_1 + x_2 + 3x_3 &= 6\end{aligned}\tag{A4-1}$$

In a *vector-matrix* format we write these equations as

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 6 \end{bmatrix}\tag{A4-2}$$

We *define* the following:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 2 & 1 & 3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} 3 \\ 1 \\ 6 \end{bmatrix}\tag{A4-3}$$

Then (A4-2) may be expressed as

$$\mathbf{Ax} = \mathbf{u}\tag{A4-4}$$

In this equation \mathbf{A} is a 3×3 (3 rows, 3 columns) *matrix*, \mathbf{x} is a 3×1 *matrix*, and \mathbf{u} is a 3×1 *matrix*. Usually matrices that contain only one row or only one column

are called *vectors*. A matrix of only one row and one column is a *scalar*. In (A4-1), x_1 is a scalar.

The general matrix A is written as

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = [a_{ij}] \quad (\text{A4-5})$$

This matrix has m rows and n columns, and thus is an $m \times n$ matrix. The element a_{ij} is the element common to the i th row and j th column.

Some special definitions are now given.

Identity matrix. The identity matrix is an $n \times n$ (square) matrix with all main diagonal elements equal to 1 and all off-diagonal elements equal to zero. For example, the 3×3 identity matrix is

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A4-6})$$

If A is also $n \times n$,

$$AI = IA = A \quad (\text{A4-7})$$

where the multiplication of matrices is defined below.

Diagonal matrix. A diagonal matrix is an $n \times n$ matrix with all off-diagonal elements equal to zero.

$$D = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \quad (\text{A4-8})$$

Symmetric matrix. The square matrix A is symmetric if $a_{ij} = a_{ji}$, for all i and j .

Transpose of a matrix. To take the transpose of a matrix, interchange rows and columns. For example,

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 2 & 1 & 3 \end{bmatrix}, \quad A^T = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 1 & 1 \\ 1 & -1 & 3 \end{bmatrix} \quad (\text{A4-9})$$

where A^T denotes the transpose of A . A property of the transpose is

$$(AB)^T = B^T A^T \quad (\text{A4-10})$$

Given the partitioned matrix

$$H = \begin{bmatrix} D & E \\ F & G \end{bmatrix} \quad (\text{A4-11})$$

where \mathbf{D} , \mathbf{E} , \mathbf{F} , and \mathbf{G} are each $n \times n$. Then \mathbf{H} is $2n \times 2n$.

$$\mathbf{H}^T = \begin{bmatrix} \mathbf{D}^T & \mathbf{F}^T \\ \mathbf{E}^T & \mathbf{G}^T \end{bmatrix} \quad (\text{A4-12})$$

Trace. The trace of a matrix is equal to the sum of its diagonal elements. Given an $n \times n$ matrix \mathbf{A} ,

$$\text{trace of } \mathbf{A} = \text{tr } \mathbf{A} = a_{11} + a_{22} + \cdots + a_{nn} \quad (\text{A4-13})$$

Eigenvalues. The *eigenvalues* (characteristic values) of a square matrix \mathbf{A} are the roots of the polynomial equation

$$|\lambda \mathbf{I} - \mathbf{A}| = 0 \quad (\text{A4-14})$$

where $|\cdot|$ denotes the determinant, and λ is a scalar.

Eigenvectors. The *eigenvectors* (characteristic vectors) of a square matrix \mathbf{A} are the vectors \mathbf{x}_i that satisfy the equation

$$\lambda_i \mathbf{x}_i = \mathbf{A} \mathbf{x}_i \quad (\text{A4-15})$$

where λ_i are the eigenvalues of \mathbf{A} .

Properties. Two properties of an $n \times n$ matrix \mathbf{A} are

$$|\mathbf{A}| = \prod_{i=1}^n \lambda_i \quad (\text{A4-16})$$

$$\text{tr } \mathbf{A} = \sum_{i=1}^n \lambda_i \quad (\text{A4-17})$$

Given the partitioned $2n \times 2n$ matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{D} & \mathbf{E} \\ \mathbf{O} & \mathbf{G} \end{bmatrix} \quad (\text{A4-18})$$

where \mathbf{D} , \mathbf{E} , and \mathbf{G} are $n \times n$, and \mathbf{O} is the $n \times n$ null matrix. Then

$$|\mathbf{H}| = |\mathbf{D}||\mathbf{G}| \quad (\text{A4-19})$$

With \mathbf{A} and \mathbf{B} square,

$$|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}| \quad (\text{A4-20})$$

Minor. The minor m_{ij} of element a_{ij} of a square matrix \mathbf{A} is the determinant of the array remaining when the i th row and j th column are deleted from \mathbf{A} . For example, m_{21} for \mathbf{A} of (A4-3) is

$$m_{21} = \begin{vmatrix} 1 & 1 \\ 1 & 3 \end{vmatrix} = 3 - 1 = 2 \quad (\text{A4-21})$$

Cofactor. The cofactor c_{ij} of element a_{ij} of the matrix \mathbf{A} is given by

$$c_{ij} = (-1)^{i+j} m_{ij} \quad (\text{A4-22})$$

For (A4-21),

$$c_{21} = (-1)^{2+1}(2) = -2 \quad (\text{A4-23})$$

Adjoint. The matrix of cofactors, when transposed, is called the adjoint of \mathbf{A} ($\text{adj } \mathbf{A}$). For \mathbf{A} of (A4-3),

$$\text{adj } \mathbf{A} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}^T = \begin{bmatrix} 4 & -5 & -1 \\ -2 & 1 & 1 \\ -2 & 2 & 0 \end{bmatrix}^T \quad (\text{A4-24})$$

Inverse. The inverse of \mathbf{A} is given by

$$\mathbf{A}^{-1} = \frac{\text{adj } \mathbf{A}}{|\mathbf{A}|} \quad (\text{A4-25})$$

where \mathbf{A}^{-1} denotes the inverse of \mathbf{A} and $|\mathbf{A}|$ denotes the determinant of \mathbf{A} . For \mathbf{A} of (A4-3) and (A4-24)

$$|\mathbf{A}| = -2$$

and

$$\mathbf{A}^{-1} = \begin{bmatrix} -2 & 1 & 1 \\ \frac{5}{2} & -\frac{1}{2} & -1 \\ \frac{1}{2} & -\frac{1}{2} & 0 \end{bmatrix} \quad (\text{A4-26})$$

Two properties of matrix inverses are

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \quad (\text{A4-27})$$

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \quad (\text{A4-28})$$

Note that the matrix inverse is defined only for a square matrix and exists only if the determinant of the matrix is nonzero. If \mathbf{A} has an inverse, so does \mathbf{A}^{-1} , with $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$. For \mathbf{A} square and $|\mathbf{A}| \neq 0$,

$$(\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1} = \mathbf{A}^{-T} \quad (\text{A4-29})$$

$$|\mathbf{A}^{-1}| = \frac{1}{|\mathbf{A}|} \quad (\text{A4-30})$$

A useful determinant. Given the partitioned matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{D} & \mathbf{E} \\ \mathbf{F} & \mathbf{G} \end{bmatrix} \quad (\text{A4-31})$$

where \mathbf{D} , \mathbf{E} , \mathbf{F} , and \mathbf{G} are each $n \times n$. Then \mathbf{H} is $2n \times 2n$. The determinant of \mathbf{H} is given by [6]

$$|\mathbf{H}| = |\mathbf{G}||\mathbf{D} - \mathbf{E}\mathbf{G}^{-1}\mathbf{F}| = |\mathbf{D}||\mathbf{G} - \mathbf{F}\mathbf{D}^{-1}\mathbf{E}| \quad (\text{A4-32})$$

provided that the indicated inverse matrices exist.

The matrix inversion lemma. The matrix inversion lemma is [6]

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \quad (A4-33)$$

provided that the indicated inverses exist.

ALGEBRA OF MATRICES

The algebra of matrices must be defined such that the operations indicated in (A4-2), and any additional operation we may wish to perform, lead us back to (A4-1).

Addition. To form the sum of matrices **A** and **B**, we add corresponding elements a_{ij} and b_{ij} , for each ij . For example,

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix} \quad (A4-34)$$

Multiplication by a scalar. To multiply a matrix **A** by a scalar k , multiply each element of **A** by k .

Multiplication of vectors. The multiplication of the $1 \times n$ (row) vector with an $n \times 1$ (column) vector is defined as

$$[x_1 x_2 \cdots x_n] \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \quad (A4-35)$$

Multiplication of matrices. An $n \times p$ matrix **A** may be multiplied by only a $p \times m$ matrix **B**; that is, the number of columns of **A** must equal the number of rows of **B**. Let

$$AB = C$$

Then the ij th element of **C** is equal to the multiplication (as vectors) of the i th row of **A** with the j th column of **B**. As an example, consider the product AA^{-1} from (A4-3) and (A4-26).

$$\begin{aligned} AA^{-1} &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} -2 & 1 & 1 \\ \frac{5}{2} & -\frac{1}{2} & -1 \\ \frac{1}{2} & -\frac{1}{2} & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I} \end{aligned} \quad (A4-36)$$

OTHER RELATIONSHIPS

Other important matrix relationships will now be given.

Differentiation. The derivative of a matrix is obtained by differentiating the matrix element by element. For example, let

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (\text{A4-37})$$

Then

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} \quad (\text{A4-38})$$

Integration. The integral of a matrix is obtained by integrating the matrix element by element. In (A4-37),

$$\int \mathbf{x} dt = \begin{bmatrix} \int x_1 dt \\ \int x_2 dt \end{bmatrix} \quad (\text{A4-39})$$

A property. Consider the scalar G :

$$G = \mathbf{x}^T \mathbf{y} = \mathbf{y}^T \mathbf{x} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \quad (\text{A4-40})$$

Then, by definition,

$$\frac{\partial G}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial G}{\partial x_1} \\ \vdots \\ \frac{\partial G}{\partial x_n} \end{bmatrix} = \mathbf{y}, \quad \frac{\partial G}{\partial \mathbf{y}} = \mathbf{x} \quad (\text{A4-41})$$

Quadratic forms. The scalar

$$F = \mathbf{x}^T \mathbf{Q} \mathbf{x} \quad (\text{A4-42})$$

is called a *quadratic form*. For example, if \mathbf{x} is second order,

$$\begin{aligned} F = \mathbf{x}^T \mathbf{Q} \mathbf{x} &= [x_1 \ x_2] \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= q_{11} x_1^2 + (q_{12} + q_{21}) x_1 x_2 + q_{22} x_2^2 \end{aligned} \quad (\text{A4-43})$$

Hence

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j \quad (\text{A4-44})$$

Note that \mathbf{Q} can be assumed symmetric with no loss of generality. Now

$$\frac{\partial F}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2q_{11}x_1 + (q_{12} + q_{21})x_2 \\ (q_{12} + q_{21})x_1 + 2q_{22}x_2 \end{bmatrix} = 2\mathbf{Q}\mathbf{x} \quad (\text{A4-45})$$

Bilinear forms. The scalar

$$G = \mathbf{x}^T \mathbf{Q} \mathbf{y} \quad (\text{A4-46})$$

is called a *bilinear form*. For example, for \mathbf{x} and \mathbf{y} second order,

$$\begin{aligned} G = \mathbf{x}^T \mathbf{Q} \mathbf{y} &= [x_1 \quad x_2] \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ &= q_{11}x_1y_1 + q_{12}x_1y_2 + q_{21}x_2y_1 + q_{22}x_2y_2 \end{aligned} \quad (\text{A4-47})$$

Hence

$$\mathbf{x}^T \mathbf{Q} \mathbf{y} = \sum_{i=1}^n \sum_{j=1}^n q_{ij}x_iy_j \quad (\text{A4-48})$$

Note that \mathbf{Q} cannot be assumed symmetric. Then

$$\frac{\partial G}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial G}{\partial x_1} \\ \frac{\partial G}{\partial x_2} \end{bmatrix} = \begin{bmatrix} q_{11}y_1 + q_{12}y_2 \\ q_{21}y_1 + q_{22}y_2 \end{bmatrix} = \mathbf{Q}\mathbf{y} \quad (\text{A4-49})$$

and

$$\frac{\partial G}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial G}{\partial y_1} \\ \frac{\partial G}{\partial y_2} \end{bmatrix} = \begin{bmatrix} q_{11}x_1 + q_{21}x_2 \\ q_{12}x_1 + q_{22}x_2 \end{bmatrix} = \mathbf{Q}^T \mathbf{x} \quad (\text{A4-50})$$

Note that these relationships are directly evident from (A4-40) and (A4-41). For example, in (A4-40), replace \mathbf{y} with $\mathbf{Q}\mathbf{y}$, and (A4-49) is obtained from (A4-41).

Sign definiteness. If the scalar $F = \mathbf{x}^T \mathbf{Q} \mathbf{x} > 0$ for all $\mathbf{x} \neq 0$, the quadratic form is *positive definite*. If $F \geq 0$ for all \mathbf{x} , the quadratic form is *positive semidefinite*. One test for sign definiteness is given in Section 10.2. A second test is that the principal minors $\Delta_i, i = 1, 2, \dots, n$, are all positive (nonnegative) for F positive definite (positive semidefinite), where

$$\Delta_1 = q_{11}, \quad \Delta_2 = \begin{vmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{vmatrix}, \quad \dots, \quad \Delta_n = |\mathbf{Q}| \quad (\text{A4-51})$$

REFERENCES

1. F. R. Gantmacher, *Theory of Matrices*, Vols. I and II. New York: Chelsea Publishing Company, Inc., 1959.
2. P. M. DeRusso, R. J. Roy, and C. M. Close, *State Variables for Engineers*. New York: John Wiley & Sons, Inc., 1965.
3. K. Ogata, *Modern Control Engineering*, 2d ed. New York: McGraw-Hill Book Company, 1990.
4. G. Strang, *Linear Algebra and Its Applications*. New York: Academic Press, Inc., 1976.
5. G. H. Golub and C. F. Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press, 1983.
6. T. E. Fortman, "A Matrix Inversion Identity," *IEEE Trans. Autom. Control*, Vol. AC-15, p. 599, Oct. 1970.

APPENDIX V

Second-Order Module Subroutines

Presented in this appendix are subroutines in Intel 8086 assembly language for the 2D, 3D, 4D, 1X, and 2X second-order digital filter structures of Chapter 13.

2D

```
;
;
;
; OUTP_2D:   Y = A0*X + P1
; X PASSED IN AX, Y RETURNED IN AX
; LOOP COUNT IN CX
OUTP_2D:     MOV     SI, #0           ; INDEX
             LEA     DI, X           ; POINT TO X
OLP_2D:      STOW    [DI]            ; SAVE X
             IMUL    A0[SI]          ; X*A0/4 IN DX
             SAL     DX, 2           ; X*A0
             ADD     DX, P1[SI]      ; Y
             MOV     AX, DX          ; RETURN IN AX
             ADD     SI, #2          ; INDEX
             LOOP    OLP_2D         ; USE COUNT IN CX
             RET
;
;
```

```

;
; PRE_2D AND DELAY_2D ARE NOT NEEDED
PRE_2D:    RET
DELAY_2D:  RET
;
;
;
; POST_2D:  P1 = A1*X - B1*Y + P2
            P2 = A2*X - B2*Y
; LOOP COUNT IN CX
POST_2D:    LEA    SI, AI        ; COEF POINTER
            LEA    BX, X        ; POINT TO INPUTS
            MOV    DI, #0       ; INDEX
POLP_2D:    LODW                   ; A1/2
            IMUL   [BX][DI]      ; X*A1/4 IN DX
            PUSH   DX           ; SAVE
            LODW                   ; B1/2
            IMUL   2[BX][DI]    ; B1*Y/4
            POP    AX
            SUB    AX, DX       ; (A1*X-B1*Y)/4
            SAL    AX, 2        ; A1*X-B1*Y
            ADD    AX, P2[DI]   ; COMPUTE P1
            MOV    P1[DI], AX   ; STORE
            LODW                   ; A2/2
            IMUL   [DX][DI]    ; X*A2/4 IN DX
            PUSH   DX
            LODW                   ; B2/2
            IMUL   2[DX][DI]    ; Y*B2/4
            POP    AX
            SUB    AX, DX       ; P2/2=(X*A2-Y*B2)/4
            SAL    AX, 2        ; P2
            MOV    P2[DI]      ; STORE P2
            ADD    DI, #2
            LOOP   POLP_2D
            RET
;
;
;
; 2D CONSTANT STORAGE FOR N STAGES
A0:      DW    A10, ..., AN0    ; A0 FOR N STAGES
A1:      DW    A11, B11, A12, B12 ; STAGE 1 COEFS
        DW    A21, B21, A22, B22 ; STAGE 2
        .
        .
        DW    AN1, BN1, AN2, BN2 ; STAGE N
; 2D TEMPORARY STORAGE FOR N STAGES
X:      DW    (N+1)DUP(0)      ; INPUTS/OUTPUTS

```

```

P1:      DW    NDUP(0)
P2:      DW    NDUP(0)

```

3D

```

;
; OUTP_3D:   $\dot{Y} = A0 \cdot X + T3$ 
; PASS X IN AX, RETURN Y IN AX
; LOOP COUNT IN CX
OUTP_3D:  LEA    DI, X1      ; POINT TO X(k)
          MOV    SI, #0      ; INDEX
OLP_3D:   STOW   ; SAVE X, Y
          IMUL   A0[SI]      ; A0*X/4 IN DX
          SAL    DX, 2       ; A0*X
          ADD    DX, T3[SI]  ; COMPUTE Y
          MOV    AX, DX      ; RETURN Y IN AX
          ADD    SI, #2      ; INDEX
          LOOP   OLP_3D
          STOW   ; SAVE LAST Y
          RET

;
;
;
; DELAY_3D:  X(k) TO X(k-1) OR X1 TO X2, Y1 to Y2
; LOOP COUNT IN CX
DELAY_3D:  LEA    SI, X1      ; X(k)
          LEA    DI, X2      ; X(k-1)
          INC    CX          ; MOVE X AND Y VALUES
          REP
          MOVW   ; BLOCK MOVE
          RET

;
;
;
; POST_3D:   NOT NEEDED
POST_3D:   RET

;
;
;
; PRE_3D:     $T3 = A1 \cdot X1 + A2 \cdot X2 - B1 \cdot Y1 - B2 \cdot Y2$ 
; LOOP COUNT IN CX
PRE_3D:    LEA    SI, A1      ; COEF POINTER
          MOV    DI, #0      ; INDEX
PRLP_3D    LODW   ; A1/2
          IMUL   X1[DI]      ; X1*A1/4 IN DX
          MOV    BX, DX      ; PARTIAL SUM IN BX

```

```

        LODW                ; A2/2
        IMUL    X2[DI]      ; X2*A2/4 IN DX
        ADD     BX, DX      ; PARTIAL SUM
        LODW                ; B1/2
        IMUL    X1+2[DI]    ; Y1*B1/4 IN DX
        SUB     BX, DX      ; TOTAL
        LODW                ; B2/2
        IMUL    X2+2[DI]    ; Y2*B2/4
        SUB     BX, DX      ; T3/4
        SAL     BX, 2       ; T3
        MOV     T3[DI], BX  ; STORE
        ADD     DI, #2      ; INDEX
        LOOP    PRLP_3D
        RET

;
;
;
; 3D CONSTANT STORAGE FOR N STAGES
A0:      DW    A10, ..., AN0      ; A0 COEFS FOR N STAGES
A1:      DW    A11, A12, B11, B12 ; COEFS FOR STAGE 1
        DW    A21, A22, B21, B22 ; STAGE 2
        DW    :                  :
        DW    AN1, AN2  BN1, BN2  ; STAGE N
;
; 3D STORAGE FOR N STAGES
X1:      DW    (N+1)DUP(0)        ; x(k), y(k)
X2:      DW    (N+1)DUP(0)        ; x(k-1), y(k-1)
T3:      DW    NDUP(0)
;
; OUTP_4D:  R0 = X + R1; Y = A0*R0 + Q1
; PASS X IN AX, RETURN Y IN AX
; LOOP COUNT IN CX
OUTP_4D:  LEA     DI, R0          ; POINT TO R0
        MOV     SI, #0          ; INDEX
OLP_4D:   ADD     AX, R1[SI]      ; R0
        STOW                ; STORE
        IMUL    A0[SI]          ; R0*A0/4 IN DX
        SAL     DX, 2           ; R0*A0
        ADD     DX, Q1[SI]      ; Y
        MOV     AX, DX          ; RETURN IN AX
        ADD     SI, #2          ; INDEX
        LOOP    OLP_4D
        RET
;
;
;
PRE_4D:  RET; NOT NECESSARY
;

```

```

;
;
; DELAY_4D: DELAY R0(k) TO R0(k-1) OR R0 TO R1
; LOOP COUNT IN CX
DELAY_4D:  LEA     SI, R0          ; R(k)
           LEA     DI, R01        ; R(k-1)
           REP
           MOVW                    ; BLOCK MOVE
           RET
;
; POST_4D:  R1 = -B1*R0 - B2*R01
;           Q1 = A1*R0 + A2*R01
; LOOP COUNT IN CX
POST_4D:   LEA     SI, B1          ; COEF POINTER
           MOV     DI, #0          ; INDEX
POLP_4D:   LODW                    ; B1/2
           IMUL    R0[DI]          ; R0*B1/4 IN DX
           MOV     BX, DX
           LODW                    ; B2/2
           IMUL    R01[DI]         ; R01*B2/4
           ADD     BX, DX          ; -R1/4
           SAL     BX, 2           ; -R1
           NEG     BX              ; R1
           MOV     R1[DI], BX      ; STORE R1
           LODW                    ; A1/2
           IMUL    R0[DI]          ; R0*A1/4
           MOV     BX, DX
           LODW                    ; A2/2
           IMUL    R01[DI]         ; R01*A2/4
           ADD     BX, DX          ; Q1/4
           SAL     BX, 2           ; Q1
           MOV     Q1[DI], BX      ; STORE Q1
           DD      DI, #2          ; INDEX
           LOOP    POLP_4D
           RET
;
; 4D CONSTANT STORAGE FOR N STAGES
A0:        DW      A10, ..., AN0  ;
B1:        DW      B11, B12, A11, A12 ; STAGE 1 COEFS
           DW      B21, B22, A21, A22 ; STAGE 2
           DW      :                ;
           DW      BN1, BN2, AN1, AN2 ; STAGE N
; 4D TEMPORARY STORAGE FOR N STAGES
R0:        DW      NDUP(0)          ; R0(k)
R01:       DW      NDUP(0)          ; R0(k-1)
R1:        DW      NDUP(0)          ; R1(k)
Q1:        DW      NDUP(0)          ; Q1(k)

```

1X

```

;
; OUTP_1X:  Y = A0*X + S2
; PASS X IN AX, RETURN Y IN AX
; LOOP COUNT IN CX
OUTP_1X:  LEA    DI, X      ; POINT TO X
          MOV    SI, #0     ; INDEX
OLP_1X:   STOW                   ; SAVE X
          IMUL   A0[SI]     ; X*A0/4
          SAL    DX, 2      ; X*A0
          ADD    DX, S2[SI] ; Y
          MOV    AX, DX     ; RET IN AX
          ADD    SI, #2     ; INDEX
          LOOP   OLP_1X
          RET

;
;
;
PRE_1X:   RET: NOT NEEDED FOR 1X
;
;
;
; DELAY_1X:  DELAY S1(k) TO S1(k-1), S2(k) TO S2(k-1)
; LOOP COUNT IN CX
DELAY_1X: LEA    SI, S1     ; SOURCE
          LEA    DI, S11    ; DESTINATION
          ADD    CX, CX     ; DOUBLE COUNT FOR
                           ; S1 and S2
          REP
          MOVW                   ; BLOCK MOVE
          RET

;
;
;
; POST_1X:   S1 = G1*S11 - G2*S21 + G3*X
             S2 = G1*S21 + G2*S11 + G4*X
POST_1X:   LEA    SI, G1     ; COEF POINTER
          MOV    DI, #0     ; INDEX
POLP_1X:   LODW                   ; G1/2
          IMUL   S11[DI]     ; S11*G1/4
          MOV    BX, DX
          LODW                   ; G2/2
          IMUL   S21[DI]     ; S21*G2/4
          SUB    BX, DX
          LODW                   ; G3/2
          IMUL   X[DI]       ; X*G3/4
          ADD    BX, DX     ; S1/4

```

```

        SAL    BX, 2      ; S1
        MOV    S1[DI], BX ; STORE S1
        LODW                   ; G1/2
        IMUL   S21[DI]     ; S21*G1/4
        MOV    BX, DX
        LODW                   ; G2/2
        IMUL   S11[DI]     ; S11*G2/4
        ADD    BX, DX

        LODW                   ; G4/2
        IMUL   X[D1]       ; X*G4/4
        ADD    BX, DX      ; S2/4
        SAL    BX, 2      ; S2
        MOV    S2[DI], BX ; STORE S2
        ADD    DI, #2      ; INDEX
        LOOP   POLP_1X
        RET

;
;
;
; 1X CONSTANT STORAGE FOR N STAGES
A0:      DW    A10, ..., AN0
G1:      DW    G11, G12, G13, G11, G12, G14 ; STAGE 1 COEFS
        DW    G21, G22, G23, G21, G22, G24 ; STAGE 2
        DW    .
        DW    .
        DW    .
        DW    GN1, GN2, GN3, GN1, GN2, GN4 ; STAGE N
;
; 1X DATA STORAGE FOR N STAGES
X:      DW    NDUP(0)      ; INPUTS
S1:      DW    NDUP(0)      ; S1(k)
S2:      DW    NDUP(0)      ; S2(k)
S11:     DW    NDUP(0)      ; S1(k-1)
S21:     DW    NDUP(0)      ; S2(k-1)

```

2X

```

;
; OUTP_2X:  Y = A0*X + T4
; PASS X in AX, RETURN Y IN AX
; LOOP COUNT IN CX
OUTP_2X:  LEA    DI, X      ; POINT TO X
        MOV    SI, #0      ; INDEX
OLP_2X:   STOW                   ; SAVE INPUTS TO STAGES
        IMUL   A0[SI]       ; X*A0/4
        SAL    DX, 2        ; X*A0
        ADD    DX, T4[SI]    ; COMPUTE Y

```

```

MOV     AX, DX      ; RETURN IN AX
ADD     SI, #2      ; INDEX
LOOP    OLP_2X
RET

;
;
;
; DELAY_2X  T1(k) TO T1(k-1), T2(k) TO T2(k-1)
; LOOP COUNT IN CX
DELAY_2X: LEA     SI, T1
          LEA     DI, T11
          ADD     CX, CX      ; DOUBLE COUNT
          REP
          MOVW
          RET

;
;
;
; PRE_2X:   T4 = G3*T1(k-1) + G4*T2(k-1)
; LOOP COUNT IN CX
PRE_2X:   LEA     SI, G3      ; COEF POINTER
          MOV     DI, #0      ; INDEX
PRLP_2X:  LODW
          IMUL    T11[DI]     ; G3*T11/4
          MOV     BX, DX
          LODW
          IMUL    T21[DI]     ; G4*T21/4
          ADD     BX, DX      ; T4/4
          SAL     BX, 2       ; T4
          MOV     T4[DI], BX  ; STORE T3(k)
          ADD     DI, #2      ; INDEX
          LOOP    PRLP_2X
          RET

;
;
;
; POST_2X:  T1 = G1*T1(k-1) + G2*T2(k-1)
;           T2 = X + G1*T2(k-1) - G2*T1(k-1)
; LOOP COUNT IN CX
POST_2X:  LEA     SI, G1
          MOV     DI, #0      ; INDEX
POLP_2X:  LODW
          IMUL    T11[DI]     ; T11*G1/4
          MOV     BX, DX

          LODW
          IMUL    T21[DI]     ; T21*G2/4
          ADD     BX, DX      ; T1/4

```



```

        SAL    BX, 2          ; T1
        MOV    T1[DI], BX    ; STORE T1(k)
        SUB    SI, #4        ; BACK POINTER UP TO G1/2
        LODW                   ; G1/2
        IMUL   T21[DI]       ; G1*T21/4
        MOV    BX, DX
        LODW                   ; G2/2
        IMUL   T11[DI]       ; G2*T11/4
        SUB    BX, DX        ; PARTIAL SUM
        SAL    BX, 2          ; T2-X
        ADD    BX, X[DI]     ; T2
        MOV    T2[DI], BX    ; STORE T2(k)
        ADD    DI, #2        ; INDEX
        LOOP   OLP_2X
        RET

;
;
;
; 2X CONSTANT STORAGE FOR N STAGES
G1:      DW    G11, G12      ; STAGE 1
        DW    G21, G22      ; STAGE 2
        .
        .
        DW    GN1, GN2      ; STAGE N
G3:      DW    G13, G14      ; STAGE 1 COEFS
        DW    G23, G24      ; STAGE 2
        .
        .
        DW    GN3, GN4      ; STAGE N
A0:      DW    A01, ..., A0N

;
; 2X TEMPORARY STORAGE FOR N STAGES
X:       DW    NDUP(0)      ; INPUTS TO STAGES
T1:      DW    NDUP(0)      ; T1(k)
T2:      DW    NDUP(0)      ; T2(k)
T11:     DW    NDUP(0)      ; T1(k-1)
T21:     DW    NDUP(0)      ; T2(k-1)
T4:      DW    NDUP(0)      ; T4(k)

```

Control Software

In this appendix two software programs written specifically for this book are described. One program, CTRL, may be obtained free of charge using the form in the rear of this book, or from the MathWorks' FTP server. The second program, CSP, can be obtained by classroom instructors directly from the first author, and may be copied for educational purposes. Both programs will run on compatible IBM® personal computers; however, CTRL requires the student version of MATLAB® [1]. Since CTRL is written in MATLAB statements, this program should execute on any computer in which MATLAB has been installed. The DOS® statements *copy* and *print* are used in printing results.

CTRL

In MATLAB, most calculations are called by statements implemented in m-files. For example, the MATLAB statement for finding the roots of a polynomial whose coefficients are in the vector **p** is

$$\mathbf{r} = \text{roots}(\mathbf{p})$$

The elements of **p** must be in memory before this command is given, and the calculated roots are placed in the vector **r**. This statement causes the m-file `roots.m` to be interpreted and executed. For example, to find the roots of the polynomial

$$x^2 - 3x + 2 = 0$$

the following MATLAB program can be executed:

```
p = [1  -3  2]
r = roots(p)
```

The roots are then printed on the computer monitor. If this program is created as an ASCII file and stored under the name, for example, polyrts.m, the program is executed by entering the statement

```
polyrts
```

If MATLAB statements that prompt the user to enter **p** replace the data statement for **p**, the program becomes general, and no programming is required by the user. In this case, the user activates the MATLAB program by entering the word matlab. Next the user enters polyrts, and the user is then prompted for the vector **p**.

CTRL is written in m-files of the form just described. The user chooses from menus the desired calculations. The program then prompts the user for the required data. Hence the user does no programming, and in particular, does no debugging. Students then may spend the available time studying the fundamentals of digital control, rather than debugging programs.

Since CTRL is written in MATLAB statements, the user may modify any part of the program. Also, the user may add options to the program or delete any part of the program. The m-files of the program illustrate programming in MATLAB, and show procedures for programming that are not evident from the user's manual [1].

Of course, students using this book may program with the original m-files of MATLAB. However, the error statements available in MATLAB are quite limited, and even simple errors can be difficult to locate and correct.

The m-files of CTRL may be placed in any of the directories created when the student version of MATLAB is installed. The directory \MATLAB\MATLAB contains the original m-files used by MATLAB. The m-files of CTRL may be placed in the directory \MATLAB, to keep these files separate from the original m-files.

A second procedure for adding CTRL to MATLAB is to create the subdirectory \MATLAB\PN3, to receive the m-files. For this case, the path \MATLAB\PN3 must be added to the MATHPATH statement in the batch file mathlab.bat, which is in the subdirectory \MATLAB\BIN.

CTRL assumes the basic control system of Figure A7-1a for classical digital control analysis and design [Chapters 1–8 of this book]. This system has the transfer function

$$T(z) = \frac{KD(z)G(z)}{1 + KD(z)G(z)H}$$

with

$$G(z) = \mathcal{Z} \left[\frac{1 - e^{-Ts}}{s} G_p(s) \right]$$

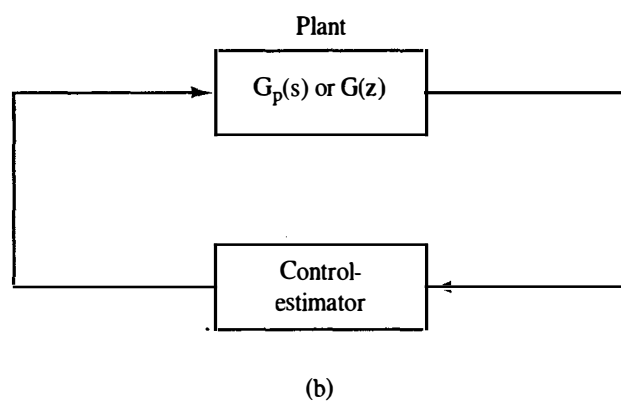
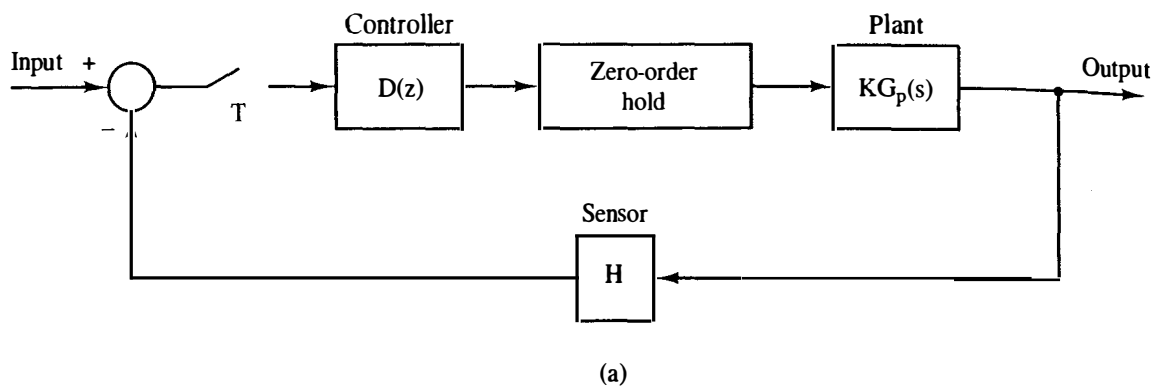


Figure A7-1. Control system configurations.

In these equations,

$D(z)$ = the controller transfer function

$G_p(s)$ = the transfer function of the analog plant

K = the constant gain that can be added to the plant

H = the *constant* gain of the sensor (no dynamics allowed)

The system of Figure A7-1b is assumed for modern digital control analysis and design [Chapters 9 and 10]. The plant has the same definition as shown previously. The controller-estimator is defined in Chapter 9.

When CTRL is first activated, the model of the plant must be entered. This operation is chosen from a menu, and the plant model may be entered as either a transfer function or a state model, and as either the analog model or the discrete model. For example, if $G_p(s)$ is entered, an analog state model is calculated. Next the transfer functions $G(z)$, $G(w)$ [see Chapter 7], and a discrete state model are calculated.

The options available in classical control (both digital and analog) are:

1. Calculate a Nyquist diagram, with or without the controller
2. Calculate a Bode diagram, with or without the controller
3. Design a phase-lead, phase lag, or a PID controller
4. Enter a controller transfer function $D(z)$
5. Calculate the closed-loop system characteristics
6. Calculate a closed-loop system time response, with or without the controller

The options available in modern digital control are:

1. Perform a pole-assignment design
2. Perform a linear-quadratic optimal design
3. Design a prediction observer
4. Design a current observer
5. Design a reduced-order observer
6. Design a Kalman filter
7. Calculate the equivalent control-observer transfer function
8. Calculate the closed-loop characteristics
9. Simulate the closed-loop system for initial-condition responses

Not all modern control options are implemented for analog control.

As stated above, no programming is required, and the user is prompted for the required data. Various types of error checking are implemented. For example, for a pole-placement-observer design, the closed-loop state model can be found if desired. Then the eigenvalues of the closed-loop system matrix is calculated to insure that the design criteria are met.

ADDITIONAL MATHEMATICAL PROCEDURES

When CTRL is first activated, the initial menu that appears is as follows:

1. Analog feedback control analysis and design
2. Digital feedback control analysis and design
3. General matrix and polynomial calculations
4. Exit program

The second option on this menu chooses the digital control programs described above. The first option chooses the analog control programs.

The third option displays the menu

1. Partial-fraction-expansion calculations
2. Transfer function to and from state space
3. Similarity transformations
4. Polynomial calculations
5. Matrix calculations
6. Exit

As an example, we consider option 5. The choice of this option displays another menu:

1. Calculate the inverse of a square matrix
2. Calculate the determinant of a square matrix
3. Calculate the eigenvalues and eigenvectors of a square matrix
4. Calculate the product of two matrices
5. Exit

These last options in general are implemented in a single MATLAB statement; the advantage in these procedures is that the user does no programming. The program prompts the user for all required data.

PRINTING RESULTS WITH MATLAB

The student version of MATLAB has no good method of printing results. Probably the easiest procedure is to use the shift-print-screen technique. This procedure dumps the screen to the printer. Generally experimentation must be used to find a procedure for dumping screen graphics to the printer. These problems are discussed in [1].

In CTRL, the *diary* statement of MATLAB is used. Diary is toggled on and off, as required, to create a data file. The *print* command of DOS is then employed to print the data file. When the user of CTRL first activates the program, the user's name is requested. This name and the time and date is then printed at the beginning of any data file, using the *copy* command of DOS. This option is useful for homework problems.

CSP

CSP is a compiled program that executes on compatible IBM personal computers. CSP has essentially the same options as CTRL. However, the additional polynomial and matrix procedures in CTRL do not appear in CSP. In addition, CSP does not have as many verifications of the calculations.

REFERENCE

1. ———, *The Student Version of MATLAB*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1992.

APPENDIX VII

The Laplace Transform

A brief review of the Laplace transform is presented in this appendix. We will see that the Laplace transform is useful in the modeling of a linear time-invariant analog system as a transfer function. The Laplace transform may also be used to solve for the response of this type of system; however, we generally use *simulations* (machine solutions of the system equations) for this purpose. For those readers wanting to delve more deeply into the Laplace transform, Refs. 1 to 3 are suggested for supplemental reading.

INTRODUCTION

By definition, the Laplace transform of a function of time $f(t)$ is [1]

$$F(s) = \mathcal{L}[f(t)] = \int_0^{\infty} f(t)e^{-st} dt \quad (\text{A8-1})$$

where \mathcal{L} indicates the Laplace transform. Note that the variable time has been integrated out of the equation and that the Laplace transform is a function of the complex variable s . The inverse Laplace transform is given by

$$f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} F(s)e^{st} ds \quad (\text{A8-2})$$

where \mathcal{L}^{-1} indicates the inverse transform and $j = \sqrt{-1}$.

Equations (A8-1) and (A8-2) form the Laplace transform pair. Given a function $f(t)$, we integrate (A8-1) to find its Laplace transform $F(s)$. Then if this function

$F(s)$ is used to evaluate (A8-2), the result will be the original value of $f(t)$. The value of σ in (A8-2) is determined by the regions of convergence of (A8-1). We seldom use (A8-2) to evaluate an inverse Laplace transform; instead we use (A8-1) to construct a table of transforms for useful time functions. Then, when possible, we use this table to find the inverse transform rather than integrating (A8-2).

As an example, we will find the Laplace transform of the exponential function e^{-at} . From (A8-1),

$$\begin{aligned} F(s) &= \int_0^{\infty} e^{-at} e^{-st} dt = \int_0^{\infty} e^{-(s+a)t} dt = \left. \frac{-e^{-(s+a)t}}{s+a} \right|_0^{\infty} \\ &= \frac{1}{s+a}, \quad \text{Re}(s+a) > 0 \end{aligned} \quad (\text{A8-3})$$

where $\text{Re}(\cdot)$ indicates the real part of the expression. Of course, Laplace transform tables were derived long ago, and we will not derive any additional transforms. Appendix VIII contains a rather extensive table of Laplace transforms and z -transforms, with the first two columns of this table giving Laplace transforms. The remaining column in this table is useful when we consider digital control systems.

From the definition of the Laplace transform, (A8-1),

$$\mathcal{L}[kf(t)] = k\mathcal{L}[f(t)] = kF(s) \quad (\text{A8-4})$$

for k constant, and

$$\mathcal{L}[f_1(t) + f_2(t)] = \mathcal{L}[f_1(t)] + \mathcal{L}[f_2(t)] = F_1(s) + F_2(s) \quad (\text{A8-5})$$

The use of these two relationships greatly extends the application of the Laplace-transform table of Appendix VIII.

We now present some examples of the Laplace transform and of the inverse Laplace transform. First, however, we need to note that using the complex inversion integral (A8-2) to evaluate the inverse Laplace transform results in $f(t) = 0$ for $t < 0$ [1]. Hence, to be consistent, we will always assign a value of zero to $f(t)$ for all negative time. Also, to simplify notation, we define the unit step function $u(x)$ to be

$$u(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (\text{A8-6})$$

In Equation (A8-3), the Laplace transform of e^{-at} was derived. Note that the Laplace transform of $e^{-at}u(t)$ is the same function. Thus for any function $f(t)$,

$$\mathcal{L}[f(t)] = \mathcal{L}[f(t)u(t)] = F(s) \quad (\text{A8-7})$$

Example A8.1

The Laplace transform of the time function

$$f(t) = 5u(t) + 3e^{-2t}$$

will now be found. From Appendix VIII and (A8-4),

$$\mathcal{L}[5u(t)] = 5\mathcal{L}[u(t)] = \frac{5}{s}$$

$$\mathcal{L}[3e^{-2t}] = 3\mathcal{L}[e^{-2t}] = \frac{3}{s+2}$$

Then, from (A8-5),

$$F(s) = \mathcal{L}[5u(t) + 3e^{-2t}] = \frac{5}{s} + \frac{3}{s+2}$$

This Laplace transform can also be expressed as

$$F(s) = \frac{5}{s} + \frac{3}{s+2} = \frac{8s+10}{s(s+2)}$$

The transforms are usually easier to manipulate in the combined form than in the sum-of-terms form.

This example illustrates an important point. As stated, we usually work with the Laplace transform expressed as a ratio of polynomials in the variable s (we call this ratio of polynomials a *rational function*). However, the tables used to find inverse transforms contain only low-order functions. Hence a method is required for converting from a general rational function to the forms that appear in the tables. This method is called the *partial-fraction expansion* method. A simple example is illustrated in the relationship

$$\frac{c}{(s+a)(s+b)} = \frac{k_1}{s+a} + \frac{k_2}{s+b}$$

Given the constants a , b , and c , the problem is to find the coefficients of the partial-fraction expansion k_1 and k_2 . We now derive the general relationships required.

Consider the general rational function

$$F(s) = \frac{b_ms^m + \cdots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0} = \frac{N(s)}{D(s)}, \quad m < n \quad (\text{A8-8})$$

where $N(s)$ is the numerator polynomial and $D(s)$ is the denominator polynomial. To perform a partial-fraction expansion, first the roots of the denominator must be found. Then $F(s)$ can be expressed as

$$F(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{\prod_{i=1}^n (s-p_i)} = \frac{k_1}{s-p_1} + \frac{k_2}{s-p_2} + \cdots + \frac{k_n}{s-p_n} \quad (\text{A8-9})$$

where Π indicates the product of terms. Suppose that we wish to calculate the coefficient k_j . We first multiply (A8-9) by the term $(s-p_j)$.

$$(s-p_j)F(s) = \frac{k_1(s-p_j)}{s-p_1} + \cdots + k_j + \cdots + \frac{k_n(s-p_j)}{s-p_n} \quad (\text{A8-10})$$

If this equation is evaluated for $s = p_j$, we see then that all terms on the right side are zero except the j th term, and thus

$$k_j = (s - p_j)F(s) \Big|_{s=p_j}, \quad j = 1, 2, \dots, n \quad (\text{A8-11})$$

In mathematics, k_j is called the *residue* of $F(s)$ in the pole at $s = p_j$.

If the denominator polynomial of $F(s)$ has repeated roots, $F(s)$ can be expanded as in the example

$$\begin{aligned} F(s) &= \frac{N(s)}{(s - p_1)(s - p_2)^r} \\ &= \frac{k_1}{s - p_1} + \frac{k_{21}}{s - p_2} + \frac{k_{22}}{(s - p_2)^2} + \dots + \frac{k_{2r}}{(s - p_2)^r} \end{aligned} \quad (\text{A8-12})$$

where it is seen that a denominator root of multiplicity r yields r terms in the partial-fraction expansion. The coefficients of the repeated-root terms are calculated from the equation

$$k_{2j} = \frac{1}{(r - j)!} \frac{d^{r-j}}{ds^{r-j}} [(s - p_2)^r F(s)] \Big|_{s=p_2} \quad (\text{A8-13})$$

This equation is given without proof [4].

The preceding development applies to complex poles as well as real poles. Consider the case that $F(s)$ has a pair of complex poles. If we let $p_1 = a - jb$ and $p_2 = a + jb$, (A8-9) can be written as

$$F(s) = \frac{k_1}{s - a + jb} + \frac{k_2}{s - a - jb} + \frac{k_3}{s - p_3} + \dots + \frac{k_n}{s - p_n} \quad (\text{A8-14})$$

The coefficients k_1 and k_2 can be evaluated using (A8-11) as before. It will be found, however, that these coefficients are complex valued, and that k_2 is the conjugate of k_1 . In order to achieve a convenient form for the inverse transform, we will use the following approach. From (A8-11),

$$\begin{aligned} k_1 &= (s - a + jb)F(s) \Big|_{s=a-jb} = Re^{j\theta} \\ k_2 &= (s - a - jb)F(s) \Big|_{s=a+jb} = Re^{-j\theta} = k_1^* \end{aligned} \quad (\text{A8-15})$$

where the asterisk indicates the conjugate of the complex number. Define $f_1(t)$ as the inverse transform of the first two terms of (A8-14). Hence

$$\begin{aligned} f_1(t) &= Re^{j\theta} e^{(a-jb)t} + Re^{-j\theta} e^{(a+jb)t} \\ &= 2Re^{at} \left[\frac{e^{j(bt-\theta)} + e^{-j(bt-\theta)}}{2} \right] \\ &= 2Re^{at} \cos(bt - \theta) \end{aligned} \quad (\text{A8-16})$$

by Euler's identity [2]. This approach expresses the inverse transform in a convenient form and the calculations are relatively simple. The damped sinusoid has an ampli-

tude of $2R$ and a phase angle of θ , where R and θ are defined in (A8-15). Three examples of finding the inverse Laplace transform are given next.

Example A8.2

In this example the inverse Laplace transform of a rational function is found.

$$F(s) = \frac{5}{s^2 + 3s + 2} = \frac{5}{(s+1)(s+2)}$$

First the partial fractional expansion is derived:

$$F(s) = \frac{5}{(s+1)(s+2)} = \frac{k_1}{s+1} + \frac{k_2}{s+2}$$

The coefficients in the partial-fraction expansion are calculated from (A8-11):

$$k_1 = (s+1)F(s) \Big|_{s=-1} = \frac{5}{s+2} \Big|_{s=-1} = 5$$

$$k_2 = (s+2)F(s) \Big|_{s=-2} = \frac{5}{s+1} \Big|_{s=-2} = -5$$

Thus the partial-fraction expansion is

$$\frac{5}{(s+1)(s+2)} = \frac{5}{s+1} + \frac{-5}{s+2}$$

This expansion can be verified by recombining the terms on the right side to yield the left side of the equation. The inverse transform of $F(s)$ is then

$$\mathcal{L}^{-1}[F(s)] = (5e^{-t} - 5e^{-2t})u(t)$$

The function $u(t)$ is often omitted, but we must then understand that the inverse transform can be nonzero only for positive time and must be zero for negative time.

Example A8.3

As a second example of finding the inverse Laplace transform, consider the function

$$F(s) = \frac{2s+3}{s^3+2s^2+s} = \frac{2s+3}{s(s+1)^2} = \frac{k_1}{s} + \frac{k_{21}}{s+1} + \frac{k_{22}}{(s+1)^2}$$

The coefficients k_1 and k_{22} can easily be evaluated:

$$k_1 = sF(s) \Big|_{s=0} = \frac{2s+3}{(s+1)^2} \Big|_{s=0} = 3$$

$$k_{22} = (s+1)^2 F(s) \Big|_{s=-1} = \frac{2s+3}{s} \Big|_{s=-1} = -1$$

We use (A8-13) to find k_{21} :

$$\begin{aligned} k_{21} &= \frac{1}{(2-1)!} \frac{d}{ds} [(s+1)^2 F(s)] \Big|_{s=-1} = \frac{d}{ds} \left[\frac{2s+3}{s} \right] \Big|_{s=-1} \\ &= \frac{s(2) - (2s+3)(1)}{s^2} \Big|_{s=-1} = \frac{-2-1}{1} = -3 \end{aligned}$$

Thus the partial-fraction expansion yields

$$F(s) = \frac{2s + 3}{s(s + 1)^2} = \frac{3}{s} + \frac{-3}{s + 1} + \frac{-1}{(s + 1)^2}$$

Then, from Appendix VIII $f(t) = 3 - 3e^{-t} - te^{-t}$.

Example A8.4

To illustrate the inverse transform of a function having complex poles, consider

$$\begin{aligned} F(s) &= \frac{10}{s^3 + 4s^2 + 9s + 10} = \frac{10}{(s + 2)(s^2 + 2s + 5)} = \frac{10}{(s + 2)[(s + 1)^2 + 2^2]} \\ &= \frac{k_1}{s + 2} + \frac{k_2}{s + 1 + j2} + \frac{k_2^*}{s + 1 - j2} \\ &= \frac{k_1}{s + p_1} + \frac{k_2}{s + p_2} + \frac{k_2^*}{s + p_2^*} \end{aligned}$$

Evaluating the coefficient k_1 as before,

$$k_1 = (s + 2)F(s) \Big|_{s = -2} = \frac{10}{(s + 1)^2 + 4} \Big|_{s = -2} = \frac{10}{5} = 2$$

Coefficient k_2 is calculated from (A8-15).

$$\begin{aligned} k_2 &= (s + 1 + j2)F(s) = \frac{10}{(s + 2)(s + 1 - j2)} \Big|_{s = -1 - j2} \\ &= \frac{10}{(-1 - j2 + 2)(-1 - j2 + 1 - j2)} = \frac{10}{(1 - j2)(-j4)} \\ &= \frac{10}{(2.236 \angle -63.4^\circ)(4 \angle -90^\circ)} = 1.118 \angle 153.4^\circ = R \angle \theta \end{aligned}$$

Therefore, using (A8-16),

$$f(t) = 2e^{-2t} + 2.236e^{-t} \cos(2t - 153.4^\circ)$$

PROPERTIES OF THE LAPLACE TRANSFORM

The Laplace transform was defined in the last section. For the analysis and design of control systems, however, we require several properties of the Laplace transform. As an example, we derive the final-value property.

Suppose that we wish to calculate the final value of $f(t)$, that is, $\lim_{t \rightarrow \infty} f(t)$. However, we wish to calculate this final value directly from the Laplace transform $F(s)$ without finding the inverse Laplace transform. The final-value property allows us to do this. To derive this property, it is first necessary to find the Laplace transform of the derivative of a general function $f(t)$.

$$\mathcal{L}\left[\frac{df}{dt}\right] = \int_0^\infty e^{-st} \frac{df}{dt} dt \quad (\text{A8-17})$$

This expression can be integrated by parts, with

$$u = e^{-st}, \quad dv = \frac{df}{dt} dt$$

Thus

$$\begin{aligned} \mathcal{L}\left[\frac{df}{dt}\right] &= uv \Big|_0^\infty - \int_0^\infty v du = f(t)e^{-st} \Big|_0^\infty + s \int_0^\infty e^{-st} f(t) dt \\ &= 0 - f(0) + sF(s) = sF(s) - f(0) \end{aligned} \quad (\text{A8-18})$$

To be mathematically correct, the initial-condition term should be $f(0^+)$ [1], where

$$f(0^+) = \lim_{t \rightarrow 0} f(t), \quad t > 0 \quad (\text{A8-19})$$

However, we will use the notation $f(0)$.

Now the final-value property can be derived. From (A8-17),

$$\begin{aligned} \lim_{s \rightarrow 0} \left[\mathcal{L}\left(\frac{df}{dt}\right) \right] &= \lim_{s \rightarrow 0} \int_0^\infty e^{-st} \frac{df}{dt} dt \\ &= \int_0^\infty \frac{df}{dt} dt = \lim_{t \rightarrow \infty} f(t) - f(0) \end{aligned} \quad (\text{A8-20})$$

Then, from (A8-18) and (A8-20),

$$\lim_{t \rightarrow \infty} f(t) - f(0) = \lim_{s \rightarrow 0} [sF(s) - f(0)] \quad (\text{A8-21})$$

or,

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s) \quad (\text{A8-22})$$

provided that the limit on the left side of this relationship exists. The right-side limit may exist without the existence of the left-side limit. For this case, the right side of (A8-22) gives the incorrect value for the final value of $f(t)$.

Table A8.1 lists several useful properties of the Laplace transform. No further proofs of these properties are given here; interested readers should see [3,4]. An example of the use of these properties is given next.

Example A8.5

As an example of applying the properties, consider the time function $\cos at$.

$$F(s) = \mathcal{L}[f(t)] = \mathcal{L}[\cos at] = \frac{s}{s^2 + a^2}$$

Then, from Table A8.1,

$$\mathcal{L}\left[\frac{df}{dt}\right] = \mathcal{L}[-a \sin at] = sF(s) - f(0) = \frac{s^2}{s^2 + a^2} - 1 = \frac{-a^2}{s^2 + a^2}$$

which agrees with the transform from Appendix VIII. Also,

$$\mathcal{L}\left(\int_0^t f(\tau) d\tau\right) = \mathcal{L}\left(\frac{\sin at}{a}\right) = \frac{F(s)}{s} = \frac{1}{s^2 + a^2}$$

TABLE A8-1 LAPLACE TRANSFORM PROPERTIES

Name	Theorem
Derivative	$\mathcal{L}\left[\frac{df}{dt}\right] = sF(s) - f(0^+)$
n th-order derivative	$\mathcal{L}\left[\frac{d^n f}{dt^n}\right] = s^n F(s) - s^{n-1}f(0^+) - \dots - f^{(n-1)}(0^+)$
Integral	$\mathcal{L}\left[\int_0^t f(\tau) d\tau\right] = \frac{F(s)}{s}$
Shifting	$\mathcal{L}[f(t - t_0)u(t - t_0)] = e^{-t_0 s} F(s)$
Initial value	$\lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} sF(s)$
Final value	$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$
Frequency shift	$\mathcal{L}[e^{-at} f(t)] = F(s + a)$
Convolution integral	$\mathcal{L}^{-1}[F_1(s)F_2(s)] = \int_0^t f_1(t - \tau)f_2(\tau) d\tau$ $= \int_0^t f_1(\tau)f_2(t - \tau) d\tau$

which also agrees with Appendix VIII. The initial value of $f(t)$ is

$$f(0) = \lim_{s \rightarrow \infty} sF(s) = \lim_{s \rightarrow \infty} \left[\frac{s^2}{s^2 + a^2} \right] = 1$$

which, of course, is correct. If we carelessly apply the final-value property, we obtain

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s) = \lim_{s \rightarrow 0} \left[\frac{s^2}{s^2 + a^2} \right] = 0$$

which is incorrect, since $\cos at$ does not have a final value; the function continues to vary between 1 and -1 as time increases without bound. This exercise emphasizes the point that the final-value property does not apply to functions that have no final value.

Example A8.6

As a second example, we consider the time function $f(t) = e^{-0.5t}$, which is then delayed by 4 s. Thus the function that we consider is

$$f_1(t) = f(t - 4)u(t - 4) = e^{-0.5(t - 4)}u(t - 4)$$

Both $f(t)$ and $f_1(t)$ are shown in Figure A8.1. Note that $f(t)$ is delayed by 4 s and that the value of the delayed function is zero for time less than 4 s (the amount of the delay). Both of these conditions are necessary in order to apply the shifting property of Table A8.1. From this property

$$\mathcal{L}[f(t - t_0)u(t - t_0)] = e^{-t_0 s} F(s), \quad F(s) = \mathcal{L}[f(t)]$$

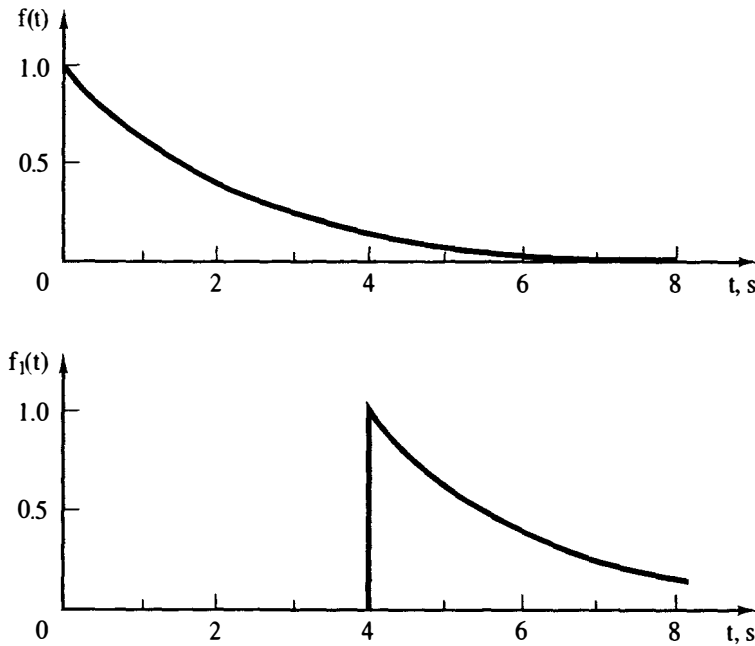


Figure A8-1 Delayed time function.

For this example, the unshifted function is $e^{-0.5t}$, and thus $F(s) = 1/(s + 0.5)$. Hence

$$\mathcal{L}[e^{-0.5(t-4)}u(t-4)] = \frac{e^{-4s}}{s + 0.5}$$

Note that for the case that the time function is delayed, the Laplace transform is not a ratio of polynomials in s but contains the exponential function.

DIFFERENTIAL EQUATIONS AND TRANSFER FUNCTIONS

In control system analysis and design, the Laplace transform is used to transform constant-coefficient linear differential equations into algebraic equations. The algebraic equations are much easier to manipulate and analyze, simplifying the analysis of the differential equations. We generally model analog physical systems with linear differential equations with constant coefficients when possible (when the system can be accurately modeled by these equations). Thus the Laplace transform simplifies the analysis and design of analog linear systems.

An example of a linear differential equation modeling a physical phenomenon is Newton's law,

$$M \frac{d^2 x(t)}{dt^2} = f(t) \quad (\text{A8-23})$$

where $f(t)$ is the force applied to a mass M , with the resulting displacement $x(t)$. It is assumed that the units in (A8-23) are consistent. Assume that we know the mass M and the applied force $f(t)$. The Laplace transform of (A8-23) is, from Table A8-2,

$$M[s^2 X(s) - sx(0) - \dot{x}(0)] = F(s) \quad (\text{A8-24})$$

where $\dot{x}(t)$ denotes the derivative of $x(t)$. Thus to solve for the displacement of the mass, we must know the applied force, the initial displacement, $x(0)$, and the initial velocity, $\dot{x}(0)$. Then we can solve this equation for $X(s)$ and take the inverse Laplace transform to find the displacement $x(t)$. We now solve for $X(s)$:

$$X(s) = \frac{F(s)}{Ms^2} + \frac{x(0)}{s} + \frac{\dot{x}(0)}{s^2} \quad (\text{A8-25})$$

For example, suppose that the applied force $f(t)$ is zero. Then the inverse transform of (A8-25) is

$$x(t) = x(0) + \dot{x}(0)t, \quad t \geq 0 \quad (\text{A8-26})$$

If the initial velocity, $\dot{x}(0)$, is also zero, the mass will remain at its initial position $x(0)$. If the initial velocity is not zero, the displacement of the mass will increase at a constant rate equal to that initial velocity.

Note that if the initial conditions are all zero, (A8-25) becomes

$$X(s) = \frac{1}{Ms^2} F(s) \quad (\text{A8-27})$$

Consider a physical phenomenon (system) that can be modeled by a linear differential equation with constant coefficients. The Laplace transform of the response (output) of this system can be expressed as the product of the Laplace transform of the forcing function (input) times a function of s (provided all initial conditions are zero), which we call the *transfer function*. We usually denote the transfer function by $G(s)$; for a mass, we see from (A8-27) that the transfer function is

$$G(s) = \frac{1}{Ms^2} \quad (\text{A8-28})$$

An example is now given.

Example A8.7

Suppose that a system is modeled by the differential equation

$$\frac{d^2 x(t)}{dt^2} + 3 \frac{dx(t)}{dt} + 2x(t) = 2f(t)$$

In this equation, $f(t)$ is the forcing function, or the input, and $x(t)$ is the response function (output). If we take the Laplace transform of this equation, we have

$$s^2 X(s) - sx(0) - \dot{x}(0) + 3[sX(s) - x(0)] + 2X(s) = 2F(s)$$

Solving this equation for the response $X(s)$,

$$X(s) = \frac{2F(s) + (s+3)x(0) + \dot{x}(0)}{s^2 + 3s + 2}$$

The transfer function is obtained by ignoring initial conditions.

$$G(s) = \frac{X(s)}{F(s)} = \frac{2}{s^2 + 3s + 2}$$

Suppose that we wish to find the response with no initial conditions and with the system input equal to a unit-step function. Then $F(s) = 1/s$, and

$$X(s) = G(s)F(s) = \left[\frac{2}{s^2 + 3s + 2} \right] \left[\frac{1}{s} \right]$$

or

$$X(s) = \frac{2}{s(s+1)(s+2)} = \frac{1}{s} + \frac{-2}{s+1} + \frac{1}{s+2}$$

by partial-fraction expansion. The inverse transform of this expression is then

$$x(t) = 1 - 2e^{-t} + e^{-2t}, \quad t \geq 0$$

Note that after a very long time, $x(t)$ is approximately unity. The final-value property yields this same result:

$$\lim_{t \rightarrow \infty} x(t) = \lim_{s \rightarrow 0} sX(s) = \lim_{s \rightarrow 0} \frac{2}{s^2 + 3s + 2} = 1$$

In the last example, the response $X(s)$ can be expressed as

$$X(s) = G(s)F(s) + \frac{(s+3)x(0) + \dot{x}(0)}{s^2 + 3s + 2} = X_f(s) + X_{ic}(s) \quad (\text{A8-29})$$

The term $X_f(s)$ is the *forced* (also called the *zero-state*) *response*, and the term $X_{ic}(s)$ is the *initial-condition* (*zero-input*) *response*. This result is general. We see then that the total response is the sum of two terms. The forcing-function term is independent of the initial conditions, and the initial-condition term is independent of the forcing function. This characteristic is a property of linear equations.

The concept of a transfer function is basic to the study of linear feedback control systems. To generalize the results of the preceding paragraphs, let a system having an output $c(t)$ and an input $r(t)$ be described by the n th-order differential equation

$$\begin{aligned} \frac{d^n c}{dt^n} + a_{n-1} \frac{d^{n-1} c}{dt^{n-1}} + \cdots + a_1 \frac{dc}{dt} + a_0 c \\ = b_m \frac{d^m r}{dt^m} + b_{m-1} \frac{d^{m-1} r}{dt^{m-1}} + \cdots + b_1 \frac{dr}{dt} + b_0 r \end{aligned} \quad (\text{A8-30})$$

If we ignore all initial conditions, the Laplace transform of (A8-30) yields

$$\begin{aligned} (s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0)C(s) \\ = (b_ms^m + b_{m-1}s^{m-1} + \cdots + b_1s + b_0)R(s) \end{aligned} \quad (\text{A8-31})$$

Ignoring the initial conditions allows us to solve for $C(s)/R(s)$ as a rational function of s , namely,

$$\frac{C(s)}{R(s)} = \frac{b_ms^m + b_{m-1}s^{m-1} + \cdots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0} \quad (\text{A8-32})$$

Note that the denominator polynomial of (A8-32) is the coefficient of $C(s)$ in (A8-31). The reader will recall from studying classical methods for solving linear differential equations that this same polynomial set equal to zero is the *characteristic equation* of the differential equation (A8-30).

Since most of the physical systems that we encountered were described by differential equations, we frequently referred to the *characteristic equation* of the system, meaning, of course, the characteristic equation of the differential equation that described the system. The a_i coefficients in (A8-30) are parameters of the physical system described by the differential equation, such as mass, friction coefficient, spring constant, inductance, and resistance. It follows, therefore, that the characteristic equation does indeed *characterize* the system, since its roots are dependent only upon the system parameters; these roots determine that portion of the system's response (solution) whose form does not depend upon the form of input $r(t)$. This part of the solution is, of course, the complementary solution of the differential equation.

REFERENCES

1. G. Doetsch, *Guide to the Applications of the Laplace and z-Transforms*. New York: Van Nostrand Reinhold, 1971.
2. J. D. Irwin, *Basic Engineering Circuit Analysis*, 3d ed. New York: Macmillan Publishing Company, 1990.
3. W. Kaplan, *Operational Methods for Linear Systems*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1962.
4. R. V. Churchill, *Operational Mathematics*, 2d ed. New York: McGraw-Hill Book Company, 1972.

PROBLEMS

A8-1. Using the defining integral for the Laplace transform, (A8-1), derive the Laplace transform of (a) $f(t) = u(t - 2.5)$; (b) $f(t) = e^{-4t}$; (c) $f(t) = t$.

A8-2. (a) Use the Laplace transform tables to find the transform of each function given.

(b) Take the inverse transform of each $F(s)$ in part (a) to verify the results.

(i) $f(t) = 3te^{-t}$

(ii) $f(t) = -5 \cos t$

(iii) $f(t) = 2e^{-t} - e^{-2t}$

(iv) $f(t) = 7e^{-0.5t} \cos 3t$

(v) $f(t) = 5 \cos(4t + 30^\circ)$

(vi) $f(t) = 6e^{-2t} \sin(t - 45^\circ)$

A8-3. (a) Find the inverse Laplace transform $f(t)$ for each function given.

(b) Verify the results in part (a) by taking the Laplace transform of each $f(t)$, using the Laplace-transform tables.

(i) $F(s) = \frac{1}{s(s+1)}$

(ii) $F(s) = \frac{3}{(s+1)(s+2)}$

(iii) $F(s) = \frac{2s+1}{s^2+s-2}$

(iv) $F(s) = \frac{10s}{s^2+5s+4}$

- A8-4.** (a) Find the inverse Laplace transform $f(t)$ for each function given.
 (b) Verify the results in part (a) by taking the Laplace transform of each $f(t)$, using the Laplace-transform tables.

$$(i) F(s) = \frac{5}{s(s+1)(s+2)}$$

$$(ii) F(s) = \frac{1}{s^2(s+1)}$$

$$(iii) F(s) = \frac{2s+1}{s^2+2s+5}$$

$$(iv) F(s) = \frac{s-30}{s(s^2+4s+29)}$$

- A8-5.** Given the Laplace transform

$$F(s) = \frac{s+5}{s^2+4s+13}$$

- (a) Express the inverse transform as a sum of two complex exponential functions.
 (b) Using Euler's relation, manipulate the result in part (a) into the form $f(t) = B\epsilon^{-at} \sin(bt + \phi)$.
 (c) Express the inverse transform as $f(t) = A\epsilon^{-at} \cos(bt + \theta)$.
 (d) Take the Laplace transform of $f(t)$ in part (c) to verify your result.
- A8-6.** (a) Plot $f(t)$ if its Laplace transform is given by

$$F(s) = \frac{\epsilon^{-t_1 s} - \epsilon^{-t_2 s}}{s}, \quad t_2 > t_1$$

- (b) The time function in part (a) is a rectangular pulse. Find the Laplace transform of the triangular pulse shown in Figure PA8-6.

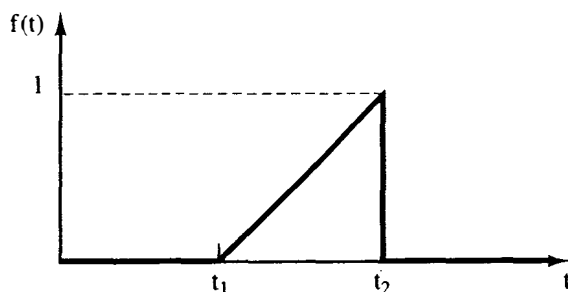


Figure PA8-6 Triangular pulse.

- A8-7.** Given that $f(t) = 4\epsilon^{-2(t-3)}$.

- (a) Find $\mathcal{L}[df(t)/dt]$ by differentiating $f(t)$ and then using the Laplace transform tables.
 (b) Find $\mathcal{L}[df(t)/dt]$ by finding $F(s)$ and using the differentiation property.
 (c) Repeat parts (a) and (b) for $f(t) = 4\epsilon^{-2(t-3)}u(t-3)$.

- A8-8.** The Laplace transform of a function $f(t)$ is given by

$$F(s) = \frac{3s+4}{s^2+3s+2}$$

- (a) Without first solving for $f(t)$, find $df(t)/dt$.
 (b) Without first solving for $f(t)$, find $\int_0^t f(\tau) d\tau$.
 (c) Verify the results of parts (a) and (b) by first solving for $f(t)$ and then performing the indicated operations.

A8-9. For the functions of Problem A8-4:

- (a) Which of the inverse transforms do not have final values; that is, for which of the inverse transforms do the $\lim_{t \rightarrow \infty} f(t)$ not exist?
- (b) Find the final values for those functions that have final values.
- (c) Find the inverse transform $f(t)$ for each function in part (b) and verify your result.

A8-10. Given $f_1(t) = u(t)$ and $f_2(t) = \sin 10t$.

- (a) Find $\mathcal{L}[f_1(t)]\mathcal{L}[f_2(t)]$.
- (b) Find $\mathcal{L}[f_1(t)f_2(t)]$.
- (c) Is $\mathcal{L}[f_1(t)]\mathcal{L}[f_2(t)]$ equal to $\mathcal{L}[f_1(t)f_2(t)]$?
- (d) Use the convolution integral of Table A8-1 to find the inverse transform of the results in part (a).
- (e) Verify the results of part (d) by finding $\mathcal{L}^{-1}[F_1(s)F_2(s)]$ directly.

A8-11. Given the differential equation

$$\frac{d^2 x(t)}{dt^2} + 5 \frac{dx(t)}{dt} + 4x(t) = 10u(t)$$

- (a) Find $x(t)$ for the case that the initial conditions are zero. Show that your solution yields the correct initial conditions; that is, solve for $x(0)$ and $\dot{x}(0)$ using your solution.
- (b) Show that your solution in part (a) satisfies the differential equation by direct substitution.
- (c) Find $x(t)$ for the case that $x(0) = 1$ and $\dot{x}(0) = 1$. Show that your solution yields the correct initial conditions; that is solve for $x(0)$ and $\dot{x}(0)$ using your solution.
- (d) Show that your solution in part (c) satisfies the differential equation by direct substitution.
- (e) Verify all partial-fraction expansions by computer.

A8-12. Given the differential equation

$$\frac{d^2 x(t)}{dt^2} + 2 \frac{dx(t)}{dt} + x(t) = 5, \quad t \geq 0$$

- (a) Find $x(t)$ for the case that the initial conditions are zero. Show that your solution yields the correct initial conditions; that is, solve for $x(0)$ and $\dot{x}(0)$ using your solution.
- (b) Show that your solution in part (a) satisfies the differential equation by direct substitution.
- (c) Find $x(t)$ for the case that $x(0) = 1$ and $\dot{x}(0) = 2$. Show that your solution yields the correct initial conditions; that is, solve for $x(0)$ and $\dot{x}(0)$ using your solution.
- (d) Show that your solution in part (c) satisfies the differential equation by direct substitution.
- (e) Verify all partial-fraction expansions by computer.

A8-13. Find the transfer function $C(s)/R(s)$ for each of the systems described by the given differential equation, where $\ddot{c}(t)$ denotes the second derivative of $c(t)$ with respect to t , and so on.

- (a) $\dot{c}(t) + 2c(t) = r(t)$
- (b) $\ddot{c}(t) + 2\dot{c}(t) = r(t - t_0)u(t - t_0) + 3\dot{r}(t)$
- (c) $\ddot{c}(t) + 3\dot{c}(t) + 2c(t) + c(t) = \dot{r}(t) + 3r(t)$

A8-14. For each of the systems, find the system differential equation if the transfer function $G(s) = C(s)/R(s)$ is given by

(a) $G(s) = \frac{60}{s^2 + 10s + 60}$

(b) $G(s) = \frac{3s + 20}{s^3 + 4s^2 + 8s + 20}$

(c) $G(s) = \frac{s + 1}{s^2}$

(d) $G(s) = \frac{7e^{-0.2s}}{s^2 + 6s + 32}$

A8-15. (a) Give the characteristic equation for the system of Problem A8-11.

(b) Give the characteristic equation for the system of Problem A8-12.

(c) Give the characteristic equations for the systems of Problem A8-13.

(d) Give the characteristic equations for the systems of Problem A8-14.

A8-16. Equations (A8-4) and (A8-5) illustrate the linear properties of the Laplace transform. This problem illustrates that these linear properties do not carry over to nonlinear operations.

(a) Given $f_1(t) = e^{-t}$, find $F_1(s) = \mathcal{L}[f_1(t)]$ and $\mathcal{L}[f_1^2(t)]$.

(b) In part (a), is $\mathcal{L}[f_1^2(t)] = F_1^2(s)$; that is, is the Laplace transform of a squared time function equal to the square of the Laplace transform of that function?

(c) Given $f_1(t) = e^{-t}$ and $f_2(t) = e^{-2t}$, find $F_1(s)$, $F_2(s)$, and $\mathcal{L}[f_1(t)/f_2(t)]$.

(d) In part (c), is $\mathcal{L}[f_1(t)/f_2(t)] = F_1(s)/F_2(s)$; that is, is the Laplace transform of the quotient of two time functions equal to the quotient of the Laplace transforms of these functions?

A8-17. (a) Give the terms that appear in the natural response for a system described by the differential equation in Problem A8-11.

(b) Give the terms that appear in the natural response for a system described by the differential equation in Problem A8-12.

(c) Give the terms that appear in the natural response for a system described by the transfer function in Problem A8-14(d).

APPENDIX VIII

z-Transform Tables

Laplace transform $E(s)$	Time function $e(t)$	z -Transform $E(z)$	Modified z -transform $E(z, m)$
$\frac{1}{s}$	$u(t)$	$\frac{z}{z-1}$	$\frac{1}{z-1}$
$\frac{1}{s^2}$	t	$\frac{Tz}{(z-1)^2}$	$\frac{mT}{z-1} + \frac{T}{(z-1)^2}$
$\frac{1}{s^3}$	$\frac{t^2}{2}$	$\frac{T^2 z(z+1)}{2(z-1)^3}$	$\frac{T^2}{2} \left[\frac{m^2}{z-1} + \frac{2m+1}{(z-1)^2} + \frac{2}{(z-1)^3} \right]$
$\frac{(k-1)!}{s^k}$	t^{k-1}	$\lim_{a \rightarrow 0} (-1)^{k-1} \frac{\partial^{k-1}}{\partial a^{k-1}} \left[\frac{z}{z - \epsilon^{-aT}} \right]$	$\lim_{a \rightarrow 0} (-1)^{k-1} \frac{\partial^{k-1}}{\partial a^{k-1}} \left[\frac{\epsilon^{-amT}}{z - \epsilon^{-aT}} \right]$
$\frac{1}{s+a}$	ϵ^{-at}	$\frac{z}{z - \epsilon^{-aT}}$	$\frac{\epsilon^{-amT}}{z - \epsilon^{-aT}}$
$\frac{1}{(s+a)^2}$	$t\epsilon^{-at}$	$\frac{Tz\epsilon^{-aT}}{(z - \epsilon^{-aT})^2}$	$\frac{T\epsilon^{-amT}[\epsilon^{-aT} + m(z - \epsilon^{-aT})]}{(z - \epsilon^{-aT})^2}$
$\frac{(k-1)!}{(s+a)^k}$	$t^k \epsilon^{-at}$	$(-1)^k \frac{\partial^k}{\partial a^k} \left[\frac{z}{z - \epsilon^{-aT}} \right]$	$(-1)^k \frac{\partial^k}{\partial a^k} \left[\frac{\epsilon^{-amT}}{z - \epsilon^{-aT}} \right]$
$\frac{a}{s(s+a)}$	$1 - \epsilon^{-at}$	$\frac{z(1 - \epsilon^{-aT})}{(z-1)(z - \epsilon^{-aT})}$	$\frac{1}{z-1} - \frac{\epsilon^{-amT}}{z - \epsilon^{-aT}}$
$\frac{a}{s^2(s+a)}$	$t - \frac{1 - \epsilon^{-at}}{a}$	$\frac{z[(aT - 1 + \epsilon^{-aT})z + (1 - \epsilon^{-aT} - aT\epsilon^{-aT})]}{a(z-1)^2(z - \epsilon^{-aT})}$	$\frac{T}{(z-1)^2} + \frac{amT-1}{a(z-1)} + \frac{\epsilon^{-amT}}{a(z - \epsilon^{-aT})}$
$\frac{a^2}{s(s+a)^2}$	$1 - (1+at)\epsilon^{-at}$	$\frac{z}{z-1} - \frac{z}{z - \epsilon^{-aT}} - \frac{aT\epsilon^{-aT}z}{(z - \epsilon^{-aT})^2}$	$\frac{1}{z-1} - \left[\frac{1 + amT}{z - \epsilon^{-aT}} + \frac{aT\epsilon^{-aT}}{(z - \epsilon^{-aT})^2} \right] \epsilon^{-amT}$

$$\frac{b-a}{(s+a)(s+b)}$$

$$\epsilon^{-at} - \epsilon^{-bt}$$

$$\frac{(e^{-aT} - \epsilon^{-bT})z}{(z - \epsilon^{-aT})(z - \epsilon^{-bT})}$$

$$\frac{\epsilon^{-amT}}{z - \epsilon^{-aT}} - \frac{\epsilon^{-bmT}}{z - \epsilon^{-bT}}$$

$$\frac{a}{s^2 + a^2}$$

$$\sin(at)$$

$$\frac{z \sin(aT)}{z^2 - 2z \cos(aT) + 1}$$

$$\frac{z \sin(amT) + \sin(1-m)aT}{z^2 - 2z \cos(aT) + 1}$$

$$\frac{s}{s^2 + a^2}$$

$$\cos(at)$$

$$\frac{z(z - \cos(aT))}{z^2 - 2z \cos aT + 1}$$

$$\frac{z \cos(amT) - \cos(1-m)aT}{z^2 - 2z \cos(aT) + 1}$$

$$\frac{1}{(s+a)^2 + b^2}$$

$$\frac{1}{b} \epsilon^{-at} \sin bt$$

$$\frac{1}{b} \left[\frac{z \epsilon^{-aT} \sin bT}{z^2 - 2z \epsilon^{-aT} \cos(bT) + \epsilon^{-2aT}} \right]$$

$$\frac{1}{b} \left[\frac{\epsilon^{-amT} [z \sin bmT + \epsilon^{-aT} \sin(1-m)bT]}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}} \right]$$

$$\frac{s+a}{(s+a)^2 + b^2}$$

$$\epsilon^{-at} \cos bt$$

$$\frac{z^2 - z \epsilon^{-aT} \cos bT}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}}$$

$$\frac{\epsilon^{-amT} [z \cos bmT + \epsilon^{-aT} \sin(1-m)bT]}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}}$$

$$\frac{a^2 + b^2}{s[(s+a)^2 + b^2]}$$

$$1 - \epsilon^{-at} \left(\cos bt + \frac{a}{b} \sin bt \right)$$

$$\frac{z(Az + B)}{(z-1)(z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT})}$$

$$\frac{1}{z-1}$$

$$A = 1 - \epsilon^{-aT} \left(\cos bT + \frac{a}{b} \sin bT \right)$$

$$- \frac{\epsilon^{-amT} [z \cos bmT + \epsilon^{-aT} \sin(1-m)bT]}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}}$$

$$B = \epsilon^{-2aT} + \epsilon^{-aT} \left(\frac{a}{b} \sin bT - \cos bT \right)$$

$$+ \frac{a}{b} \frac{\{\epsilon^{-amT} [z \sin bmT - \epsilon^{-aT} \sin(1-m)bT]\}}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}}$$

$$\frac{1}{s(s+a)(s+b)}$$

$$\frac{1}{ab} + \frac{\epsilon^{-at}}{a(a-b)}$$

$$\frac{(Az + B)z}{(z - \epsilon^{-aT})(z - \epsilon^{-bT})(z - 1)}$$

$$A = \frac{b(1 - \epsilon^{-aT}) - a(1 - \epsilon^{-bT})}{ab(b-a)}$$

$$+ \frac{\epsilon^{-bt}}{b(b-a)}$$

$$B = \frac{a\epsilon^{-aT}(1 - \epsilon^{-bT}) - b\epsilon^{-bT}(1 - \epsilon^{-aT})}{ab(b-a)}$$

Index

A

Accuracy, steady-state, 218–221, 282
Ackermann's formula, 344
Actuators, 1, 7
Admissible control, 383
Aircraft landing system. *See* Control systems
Algebraic loops, 154–56
Aliasing effect. *See* Frequency aliasing
Analog computers, 222
Analog filters. *See* Filters, analog
Analog-to-digital converters, 27, 113–24
 counter ramp, 114
 dual ramp, 120
 model, 139
 parallel, 122
 successive approximations, 116
 tracking, 115
Antenna pointing system. *See* Control systems
Antialiasing filter, 107, 447
ARMA model, 406

B

Bandwidth, 285
Bellman's principle, 397

Bilinear form, 389, 643
Bilinear transformation, 240
Bode diagrams, 261–64

C

Canonical forms:
 control, 57, 343
 observer, 57
Carbon dioxide control system. *See* Control systems
Cascade Compensation, 289
Case studies, 597–623
 aircraft landing system, 613
 environmental system, 605
 servomotor system, 598
Cauchy's principle of argument, 254
Causal system, 98, 261
Characteristic equation:
 matrix, 639
 state model, 64, 355
 transfer-function model, 210, 289
Characteristic values, 64, 639
Characteristic vectors, 67, 639
Classical control, 382
Closed-loop control, 1
Coefficient quantization, 541–46
 error analysis, 542
 pole/zero design, 541

Compensator, 1
 Compensation. *See* Design
 Compensator, 1
 Constant M circles, 267
 Control effort, 288, 383
 Control-estimator. *See* Transfer functions
 Controllability, 365
 Controller. *See* Compensator
 Control law, 383
 Control problem, 7
 Control software, 226, 654–59
 Control systems:
 aircraft landing, 3–5, 30, 613–22
 analog, 3, 29
 antenna pointing, 12–14, 21, 89, 168,
 198, 200, 275, 277, 279, 333
 carbon dioxide, 605
 continuous time. *See* Control systems,
 analog
 digital, 3
 discrete time, 2, 27
 missile, 195
 regulator, 369
 robotic, 14–16, 28, 26, 166, 170, 197,
 210, 229, 230, 275, 277, 279, 332,
 satellite, 24, 167, 197, 200, 230, 276,
 277, 278, 334, 378, 380, 423–24
 servo, 10, 12, 598
 temperature, 16–18, 23, 167, 169,
 170–71, 197, 200, 228, 274, 277,
 280, 331, 335, 377, 380, 425–25,
 607
 Controller design. *See* Design
 Convolution:
 complex, 632
 discrete, 47
 real, 138
 Cost function, 382
 quadratic, 384
 Costate vector, 398
 Covariance, 415, 417
 CSP controls program, 226, 654, 658
 CTRL controls program, 226, 654–58
 Curve fitting, least-squares, 404–06

D

Damping ratio ζ , 209, 216
 Data acquisition, 613
 Data holds:
 first-order, 107–10
 fractional-order, 110–111

n th-order, 103
 polygonal, 128
 zero-order, 103–07
 Data reconstruction, 102–111, 284
 dc gain, 136
 Design:
 current observer, 361–65
 IH-LQG, 416
 Kalman filter, 413
 lag-lead, 307–09
 linear quadratic optimal, 389–97
 phase-lag, 291–97
 phase-lead, 297–307
 PID, 312–19
 pole-assignment, 338–45
 with inputs, 369–74
 prediction observer, 345–57
 reduced-order observer, 357–61
 root locus, 319–27
 Design equations, 824–25
 Difference equations:
 definition, 29, 54
 transfer functions, 50, 55, 68
 solution, 37–40
 Differential equations:
 definition, 29
 transfer functions, 30, 668–71
 solution, 670
 Digital controllers, 138, 289
 Digital filter realizations:
 cascade, 475
 ladder, 485
 parallel, 478
 Digital filter structures:
 direct:
 first(1D), 82, 466
 second(2D), 469
 third(3D), 83, 469
 fourth(4D), 470
 1X, 83, 473, 499
 2X, 473, 499
 Digital filters, 29
 Digital-to-analog conversion, 111–13
 resolution, 111, 546
 Digital-to-analog converter, 27, 112
 Discrete-time systems, 27
 open-loop, 131
 state models, 150–61
 with digital filters, 138–42
 closed-loop, 173–91
 Disturbances, 4
 rejection, 7, 287
 Dynamic range, 533, 598

E

Eigenvalue. *See* Characteristic value
Eigenvector. *See* Characteristic vector
Error bounds:
 absolute upper bound, 557
 limit cycles, 567
 steady-state bounds, 557
Error constants:
 K_p , position, 219
 K_v , velocity, 220
Expectation, 415

F

Feedback compensation, 290
Feedback control, 1
Filter, 1
 alpha, 621
 alpha-beta, 619
 design. *See* Design
 realization, 493–511
Filter transformations, analog, 477
Filters, analog:
 Bessel, 450
 Butterworth, 447
 Chebyshev, 451
 elliptic, 454
 transitional, 450
Final value property:
 Laplace transform, 665
 z-transform, 36
Finite wordlength effects, 525
Flow graphs, 50
 original, 177
 sampled, 178
Foldover. *See* Frequency aliasing
Fourier transform, 97–99
Frequency aliasing, 107
Frequency response, 98–99, 284
 closed-loop, 266–71
 interpretation, 254
 open-loop, 303
Frequency spectrum, 98
Fundamental matrix. *See* State transition matrix

G

Gain margin, 259, 285
Gain-phase diagrams, 264

Gaussian distribution, 414
Generating function, 31

H

Hamiltonian, 398

I

Ideal sampler, 92
Ideal time delay, 49
Identifiable, 408
IH-LQG control, 416–18
Impulse modulator. *See* Ideal sampler
Interrupts, 611
Invariance:
 impulse, 436
 impulse integrator, 437
 step, 438
Inversion integral:
 Laplace transform, 660
 z-transform, 30, 45

J

Jury's test, 245–49

K

Kalman filters, 413–20

L

LabVIEW, 512–23
Ladder structures, 485
Laplace transform, 660–74
 tables, 676–77
 properties, 667
Least-squares:
 curve fitting, 404–06
 estimate, 405
 minimization, 420
 system identification:
 batch, 408–410
 recursive, 410–13
 weighted, 410

Limit cycles, 561–74
 absence, 571–74
 overflow oscillations, 572
 Linear quadratic optimal control, 389–97
 Linear systems. *See* Systems
 Loop gains, 628
 LQ, 389
 LQG, 415
 IH-LQG, 416–18

M

Mapping:
 s to z , 211–18
 z to w , 240
 Mapping functions:
 backward difference, 439
 bilinear z -transform, 441
 forward difference, 440
 matched z -transform, 443
 standard z -transform, 439
 Marginal stability, 236
 Mason's gain formula, 626–30
 MATLAB, 38, 45, 48, 68, 71, 73, 84, 160,
 161, 209, 395, 410, 418, 252, 264,
 302, 318, 352, 360, 364, 448–50,
 452–54, 458–61
 Matrices, review, 637–44
 Matrix:
 characteristic equation, 639
 characteristic values and vectors, 64, 67
 inverse, 640
 inversion lemma, 641
 modal, 67
 similarity transformation, 63–68
 Microcomputers, Intel 80 \times 86, 493–97
 filter implementation, 497–511
 Minimum principle, 397–98
 Missile control system. *See* Control
 systems
 Modern control, 382
 Modules, 475, 497
 cascaded, 475, 506
 modules, second-order, 470
 1D, 471, 497
 2D, 471, 506
 3D, 471, 497
 4D, 471, 499
 1X, 83, 473, 499
 2X, 474, 499
 paralleled, 478, 505

N

Natural frequency ω_n , 210, 216, 284
 Nichols chart, 270
 Nodes, 628
 Noise models, 554
 Noise variance, 528, 531, 547, 548, 553,
 558
 Nonlinearities:
 limit cycles, 561, 565
 overflow, 534, 564
 quantizers, 526
 Number systems, 525–40
 signed-magnitude, 526–34
 two's complement, 534–40
 Numerical approximations:
 backward difference, 431
 forward difference, 433
 left-side rule, 434
 rectangular rule, 434
 right-side rule, 435
 Simpson's rule, 436
 trapezoidal rule, 436
 Numerical differentiation, 82, 310
 Numerical integration:
 Euler rule, 221
 predictor-corrector, 224
 rectangular rule, 28, 81, 221
 Runge-Kutta, 226
 trapezoidal rule, 81, 224, 310
 Nyquist criterion, 252–61

O

Observability, 365
 Observer canonical form, 57
 Observers:
 current, 361
 optimal, Kalman filter, 413
 prediction, 345
 reduced-order, 357
 Open-loop function, 238–39
 Open-loop transfer function, 239
 Optimal control, 382–83
 quadratic, 392
 nonrecursive solution, 401
 steady-state, 402
 Optimal state estimation, 413
 Optimality principle, 386
 Ordering of modules, 587

Overflow, 534
Overshoot, 208, 283

P

Partial-fraction expansion, 662
Path gain, 628
Performance index. *See* Cost function
Persistently exciting, 408
Phase margin, 259, 285
Phase-variable canonical form, 57
Physical realizability, 261
PID controllers:
 analog, 28
 design. *See* Design
 digital, 312, 483
Pole assignment design. *See* Design
Pole-zero cancellation, 323
Pole-zero mapping, 211
Pole-zero pairing, 586
Positive definite, 389
Positive semidefinite, 389
Principle of optimality, 386
Probability density functions, 529, 537
Pulse transfer function. *See* Transfer functions

Q

Quadratic forms, 383–84, 389, 642
Quantization effects, 295, 525–40
 coefficients, 541–46
 signals, 546–560
Quantizers:
 least-significant-bit-1, 531
 roundoff, 530
 truncation, 527

R

Random inputs, 414
Rational function, 662
Regulator control system. *See* Control systems
Relative stability, 259, 285
Residues, 95, 633
 theorem of, 633

Ricatti equation:
 algebraic, 402
 discrete, 399
Rise time, 283
Robotic control system. *See* Control systems
Robust control, 365, 416, 418
Root Locus, 249–52
 design, 319–27
Routh-Hurwitz criterion, 242–45

S

Sampled-data systems, 89
Sampled-data transformations, 431–47
Sample period, 3
Sampler/data-hold model, 91
Sampling:
 ideal, 92
 nonsynchronous, 147–50
Sampling frequency, 95
Sampling theorem, Shannon, 101
Satellite:
 control system. *See* Control systems
 model, 9–10
Scaling:
 averaging, 584
 Lp-norm, 581
 optimization, 585
 unit-step, 583
 upper bound, 579
Sensitivity, 7, 286
Sensors, 1, 7
 inaccuracies, 4
Servomotor system model, 1–12
Servo system. *See* Control systems
Settling time, 283
Sign definiteness, 384, 643
Signal flow graphs, 50
 original, 177
 sampled, 178
Signal quantization, 503–04, 546–60
 input, 546
 internal variable, 548
 output, 554
Similarity transformations, 63–68
 properties, 65
 to a diagonal matrix, 66
Simulation, 221–26
Simulation diagrams:
 analog, 152
 discrete, 48

Specifications, 282, 337, 382
 Spectrum, amplitude, 98, 100
 Software, control systems, 654–59
 CSP, 226, 658
 CTRL, 226, 654
 Specifications, 282, 337, 382
 Spectrum, amplitude, 98, 100
 Standard deviation, 417
 Starred transform, 93, 95, 631–36
 properties, 99
 State equations:
 analog, 150
 solutions, 156
 discrete, 54,
 solutions, 71–77
 State estimation. *See* Observers
 State models:
 analog, 10, 12,
 transfer functions, 160
 closed-loop, 183–191
 discrete, 54, 76
 transfer functions, 68–71
 open-loop, 150–59
 State transition matrix:
 analog, 154
 discrete, 72
 properties, 75
 State variables:
 analog, 152–56
 discrete, 53–77
 Steady-state accuracy. *See* Accuracy
 Steady-state optimal control, 398–404
 Subroutines, second-order modules,
 501–02, 508–11, 645–53
 Superposition, 2
 System identification:
 least-squares, 406–10
 recursive, 410–13
 weighted least-squares, 410
 Systems:
 analog, 2
 causal, 98
 continuous-time, 2
 digital control, 138, 132
 discrete, 2, 27
 linear, 2
 sampled-data, 89
 open-loop, 131–61
 closed-loop, 173–91
 time-invariant, 2, 54
 time response, 202
 time-varying, 54, 384
 with time delay, 144

System type, 219

T

Taylor series, 102
 Temperature control system. *See* Control
 systems
 Thermistors, 23
 Time constant τ , 208, 216, 285
 Time delay, ideal:
 in a system, 144
 starred transform, 97
 z-transform, 50
 Time response, 202–27
 Timing, digital control, 611
 Trace of a matrix, 639
 in similarity transformations, 65
 Transfer functions:
 closed-loop, 173–83
 control-estimators, 352, 359, 363
 pulse, 113–34
 state models, 68–71, 160
 zeros, 371
 Transforms, filter design:
 backward difference, 439
 bilinear, 441, 457
 forward difference, 440
 bilinear z , 443, 457
 matched z , 443, 457
 Simpson's, 444
 standard z , 439, 455
 (w , v), 444
 z-forms, 445
 Transient response, 7, 236, 283
 Transpose networks, 467
 Triac, 606
 Two-point boundary value problem, 398

U

Unit impulse function, discrete, 260
 Unit time delay, 48

V

Variance, 414
 Vectors, 637–38

W

w-plane, 240

Z

z-transform:
definition, 30

delayed, 142
double sided, 31
inverse, 40–48
modified, 142–44
properties, 32–37
relation to Laplace transform, 131
tables, 42, 676–77